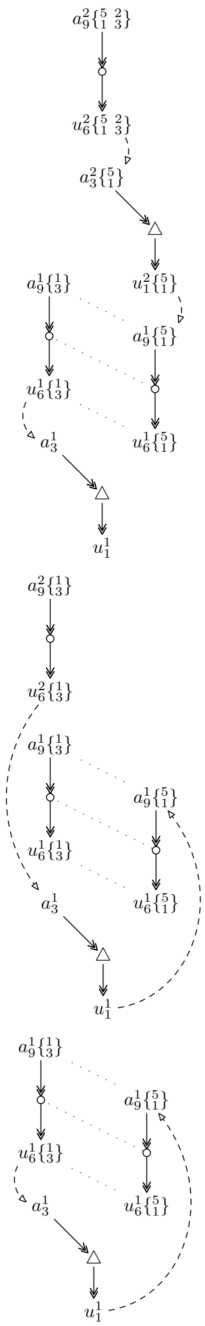
Free variable sequent calculi

one of them even with uniform variable splitting

Roger Antonsen



THESIS PRESENTED

FOR THE DEGREE OF

CANDIDATUS PHILOLOGIAE

Under supervision by:

Arild Waaler, *Department of Informatics*

Language, Logic and Information

Department of Linguistics

University of Oslo

May MMIII

Preface

This is a work in progress. To meet the requirements of a deadline and in order to finish in time, I had to give the thesis away from me at this point.

This is also a thesis about free variable sequent calculi for first-order languages without equality. A brief summary: Chapter 1 and 2 are mainly background material. Chapter 3 introduces a way of representing and reasoning about relations between inferences and sketches a method for syntactical soundness proofs. Chapter 4 investigates a new free variable calculus with *variable splitting*. There is also an appendix containing the article *"A free variable sequent calculus with uniform variable splitting"* [45] written by Arild Waaler and myself.[1] This article is a documentation of much of the work spent with this thesis.

A lot of background material can be found in this thesis. It has been my goal to write a *readable* and understandable thesis, so along with all the definitions and lemmas there are many motivating examples. Some mathematical background knowledge is recommended; from Chapter 3 and outwards some of it is also presupposed.

Scientific acknowledgments

Many ideas found in this thesis are due to Arild Waaler. In particular, the idea for a calculus with uniform variable splitting and much of the contents of the article [45], like the sketched proof of cycle elimination there, is due to him. The diagram representations have evolved through numerous conversations between Arild and me in order to understand and prove a general cycle elimination theorem. The idea for cycle elimination, as explained in Chapter 3, is basically mine. Also, most of the terminology and concepts related to uniform variable splitting *after* the article was written, is due to me.

Roger Antonsen
May 12, 2003

[1] Accepted as a research paper to the conference TABLEAUX 2003 in Rome, Italy, 9-12 September 2003.

Acknowledgments

First and foremost I want to thank my supervisor, Arild Waaler, who has generously shared both his time and his logical intuitions with me over the past two years. He has been ever-supporting and encouraging throughout the work with this material, and without doubt, I have never before met a person with such an intellectual honesty and from which I have learned so much logic. It has been a great, great experience to have had you as my supervisor, and I hope that our long conversations and explorations of logic will not end with this...

I also want to thank my internal supervisor Herman Ruge Jervell, who has given me many helpful hints and pointers during the entire period I have been a student at Language, Logic and Information.

Thanks to my parents for their unconditional support and love through all my years of studying! Many of the topics I have studied – actually, most of them – have been quite esoteric in their eyes; nevertheless, their interest and willingness to understand what I have been doing have always been present.

Special thanks goes to Tonje for prof-reading and spel-checking with the utmost scrutiny! I'm also grateful to her because of her infinite patience, understanding and caring in critical moments of discovery. Living together with a narrow-minded and obsessed wannabe-logician trying to eliminate cycles and split variables cannot be easy. Without your enduring support, this would not have been possible. Many hugs and kisses go to you.

There are two great academic environments I have been fortunate to be part of. First, I want to thank all students and teachers of Language, Logic and Information at the Department of Linguistics, residing in and around Hundremeterskogen (the Hundred Acre Woods). I first encountered this environment in the spring of 1998, and since then I have more or less been a regular inhabitant. It has been one of the reasons that my time of studying has been interesting and endurable. I hope that this will continue to be an equally great study environment in the future! Second, I want to thank all Master of Logic students at the University of Amsterdam (Institute for Language, Logic and Computation) during the study year of 2001/2002. Late night working at the study room in Plantage Muidergracht and at home in Prinsengracht will be remembered as a proof of how enjoyable and inspiring collaboration can be.

And thanks to Ludvig van Beethoven for having put the combinatorics of his piano sonata into the world and for having inspired me late at night when nobody else was around.

Contents

1 Logical preliminaries **1**
- 1.1 Syntax . . . 1
- 1.2 Semantics . . . 5
- 1.3 Sequents, rules and calculi . . . 6
- 1.4 The calculus LK . . . 12
- 1.5 Soundness of LK . . . 14
- 1.6 Completeness of LK . . . 17

2 Free variable sequent calculi **21**
- 2.1 Proof transformations . . . 24
- 2.2 The calculus LK^{δ} . . . 26
- 2.3 Soundness and completeness of LK^{δ} . . . 28
- 2.4 Formula trees . . . 30
- 2.5 The calculus $\mathsf{LK}^{\mathsf{ce}}$ . . . 33
- 2.6 Completeness of $\mathsf{LK}^{\mathsf{ce}}$ . . . 36
- 2.7 Soundness of $\mathsf{LK}^{\mathsf{ce}}$ . . . 36
  - 2.7.1 The semantical approach . . . 37
  - 2.7.2 Why local soundness does not hold for $\mathsf{LK}^{\mathsf{ce}}$ . . . 38

3 A change of perspective **45**
- 3.1 Relations between inferences . . . 45
- 3.2 Diagram representations . . . 47
- 3.3 Permutations . . . 50
- 3.4 Cycle elimination . . . 54

4 Uniform variable splitting **65**
- 4.1 Rigid and universal variables . . . 65
- 4.2 A calculus with uniform variable splitting . . . 67
- 4.3 A big example . . . 77
  - 4.3.1 Remark on non-balanced skeletons . . . 79
- 4.4 Soundness . . . 81
- 4.5 A framework for reasoning about splitting . . . 86
- 4.6 Summary and remarks . . . 89
  - 4.6.1 Ideas for further development . . . 90

Appendix **91**

A Article **93**

A.1 Introduction . . . 94
A.2 The free variable system . . . 97
A.3 Connections, colorings and proofs . . . 102
A.4 Consistency . . . 105

Chapter 1

Logical preliminaries

"Before I begin my speech,
there's something I want to say."

Saul Gorn [26]

The purpose of this chapter is to introduce the reader to the formal framework on which the subsequent chapters will depend, in particular the syntax and semantics of first-order logic and the concepts of sequents, rules and calculi.

Formulas and terms are the first notions that must be precisely defined. Intuitively, formulas are objects that can be true or false, and terms are objects that denote elements of some domain. We confine ourselves to the *first-order* case (without equality), and thus, the elements denoted by terms are individuals, not sets of individuals or sets of sets of individuals, etc. When a formula or a terms is given, there will always be an underlying *language* consisting of basic parts from which the formulas and terms are constructed.

1.1 Syntax

All languages of consideration here will have the following in common:

Quantifier symbols: $\forall$ (for all), $\exists$ (there is)

Propositional connectives: $\wedge$ (and), $\vee$ (or), $\rightarrow$ (implies), $\neg$ (not)

Variables: x_1, x_2, x_3, $\ldots$ (x, y, z, $\ldots$ will be used as abbreviations)

Auxiliary symbols: ')', '(' and ','

These are called the *logical symbols* of first-order languages. Variables will be given a special treatment later. For now, all languages will consist of the same set of variables.

The logical symbols are common for all first-order languages. Next follow the parts which will vary from language to language.

1.1 Definition (First-order language)

A *first-order language* $\mathcal{L}$ consists of:

- A countable (possibly finite) set $\mathcal{P}$ of *predicate symbols*.
- A countable (possibly finite) set $\mathcal{F}$ of *function symbols*.

Each predicate and function symbol will have associated with it a natural number, called the *arity* of the given symbol. Symbols of arity n are called n*-ary* symbols; if a symbol has arity one then it is called *unary*, and if it has arity two, then it is called *binary*. Function symbols of arity 0 are called *constant symbols*. The first-order language consisting of $\mathcal{P}$ and $\mathcal{F}$ will be denoted $\mathcal{L}(\mathcal{P}, \mathcal{F})$, or just $\mathcal{L}$ when $\mathcal{P}$ and $\mathcal{F}$ are clear from the context. $\dashv$

Before formulas of a first-order language can be defined, the terms of a first-order language must be defined, since terms are part of formulas.

1.2 Definition ($\mathcal{L}$-term)

Let $\mathcal{L}$ be a first-order language. The set of $\mathcal{L}$*-terms* is the least set that satisfies the following conditions:

- Any variable is an $\mathcal{L}$-term.
- If f is an n-ary function symbol in $\mathcal{F}$ and $t_1, \ldots, t_n$ are $\mathcal{L}$-terms, then $f(t_1, \ldots, t_n)$ is an $\mathcal{L}$-term.

An $\mathcal{L}$-term is *ground* if there are no variables in it. $\dashv$

1.3 Definition ($\mathcal{L}$-formula)

Let $\mathcal{L}$ be a first-order language. The set of $\mathcal{L}$*-formulas* is the least set that satisfies the following conditions:

- If P is an n-ary predicate symbol in $\mathcal{P}$ and $t_1, \ldots, t_n$ are $\mathcal{L}$-terms, then $P(t_1, \ldots, t_n)$ is an $\mathcal{L}$-formula. It is called an *atomic formula*.
- If φ is an $\mathcal{L}$-formula, then $\neg\varphi$ is an $\mathcal{L}$-formula.
- If φ and ψ are $\mathcal{L}$-formulas and $\circ$ is in $\{\wedge, \vee, \rightarrow\}$, then $(\varphi \circ \psi)$ is an $\mathcal{L}$-formula.

- If φ is an $\mathcal{L}$-formula and x is a variable, then $\forall x\varphi$ and $\exists x\varphi$ are $\mathcal{L}$-formulas.

When the language $\mathcal{L}$ can be understood from the context, we will skip the prefix and just speak of terms and formulas. It is nevertheless important to remember that terms and formulas are *always* given relative to some underlying first-order language $\mathcal{L}$. $\dashv$

A few conventions. Capital roman letters (P, Q, R, etc.) will be used as predicate symbols. Non-capital roman letters (f, g, h, etc.) will be used as function symbols. Capital Greek letters (φ, ψ, ξ, etc.) will be used as meta-symbols for first-order formulas.

For the rest of this section, let $\mathcal{L}$ be a fixed first-order language.

1.4 Definition (Free variable)

The *free variable occurrences* of a formula are defined recursively like this:

- The free variable occurrences of an atomic formula are all variables occurring in it.
- The free variable occurrences of $\neg\varphi$ are the free variable occurrences of φ.
- The free variable occurrences of $(\varphi \circ \psi)$, where $\circ$ is $\wedge$, $\vee$ or $\rightarrow$, are the free variable occurrences of φ together with the free variable occurrences of ψ.
- The free variable occurrences of $\forall x\varphi$ or $\exists x\varphi$ are the free variable occurrences of φ except for occurrences of x.

A variable occurrence in a formula is *bound* if is not free. $\dashv$

1.5 Definition (Closed formula)

A formula is *closed* if it has no free variable occurrences in it. $\dashv$

Example 1.6 Let $\mathcal{L}(\mathcal{P}, \mathcal{F})$ be the language consisting of the predicate symbols Q of arity one, the predicate symbol R of arity two and the function symbol s of arity one. The the following are $\mathcal{L}(\mathcal{P}, \mathcal{F})$-formulas:

$$\forall x R(z, x) \qquad \forall x \exists y (Q(x) \rightarrow (Q(z) \wedge \neg R(s(x), y)))$$

z is the only free variable occurrence in both formulas. We will often skip many of the parenthesis and write

$$\forall x Rzx \qquad \forall x \exists y (Qx \rightarrow Qz \wedge \neg Rsxy)$$

under the assumption that some connectives bind stronger than others. (We assume that $\neg$, together with quantifiers, binds stronger than $\wedge$, which binds stronger than $\vee$, which again binds stronger than $\rightarrow$.)

1.7 Definition (Substitution)

Let $\mathcal{L}$ be a first-order language. A *substitution* for $\mathcal{L}$ is a function σ from variables to $\mathcal{L}$-terms. The application of σ to an argument x can be written either like $\sigma(x)$ or $x\sigma$. It can be extended to a function from $\mathcal{L}$-terms to $\mathcal{L}$-terms in the following way:

- $c\sigma = c$
- $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$

It can be further extended to a function from $\mathcal{L}$-formulas to $\mathcal{L}$-formulas, but then we must pay attention to free and bound variable occurrences. Bound occurrences should not be substituted. Therefore, for each substitution σ, we define σ_x which is identical to σ for all arguments except x, which is left untouched. That is, for all variables y

$$\sigma_x(y) = \begin{cases} \sigma(y) & \text{if } y \neq x \\ x & \text{if } y = x \end{cases}$$

We can then proceed to define substitutions for formulas.

- For atomic formulas: $P(t_1, \ldots, t_n)\sigma = P(t_1\sigma, \ldots, t_n\sigma)$
- $(\neg\varphi)\sigma = \neg(\varphi)\sigma$
- $(\varphi \circ \psi)\sigma = (\varphi\sigma \circ \psi\sigma)$, for $\circ$ either $\wedge$, $\vee$ or $\rightarrow$
- $(Qx\varphi)\sigma = Qx(\varphi\sigma_x)$, for Q either $\forall$ or $\exists$

Often we are only interested in substitutions for a given *finite* set of variables. The *support* of a substitution σ is the set of variables x such that $x\sigma \neq x$. If the support is finite, or if we are only interested in a finite part of the substitution, we write $\{x_1/t_1, \ldots, x_n/t_n\}$ or $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ for the substitution that maps the variables $x_1, \ldots, x_n$ to $t_1, \ldots, t_n$, respectively. The result of applying such a substitution to a formula φ is written $\varphi[x_1/t_1, \ldots, x_n/t_n]$. $\dashv$

In the above definition, the situation where variables *become* bound as a result of applying a substitution is not dealt with explicitly. For the purpose of this exposition, it is not necessary. I.e., if we apply the substitution $\{z/x\}$ to the formula $\forall xPxz$, we obtain $\forall xPxx$, which is not desirable. In cases like this, we assume that bound variables have been renamed properly before applying the substitution, i.e. the formula becomes $\forall yPyz$ before applying the substitution and $\forall yPyx$ after.

1.8 Definition (Substitution composition)

Let σ and τ be substitutions The *composition* of σ and τ, denoted $\sigma\tau$, is the substitution which sends each variable x to $(x\sigma)\tau$. $\dashv$

1.2 Semantics

1.9 Definition (Structure)

Let $\mathcal{L}(\mathcal{P}, \mathcal{F})$ be a first-order language. An *$\mathcal{L}$-structure* M consists of a *domain* D and an *interpretation function* $(\cdot)^M$ such that

- D is a non-empty, countable set, also written $|M|$.
- If f is an n-ary function symbol, then $f^M : D^n \to D$.
- If R is an n-ary predicate symbol, then $R^M \subseteq D^n$.

Thus the interpretation function provides an interpretation of all the predicate and function symbols in the language. $\dashv$

1.10 Definition (Interpretation of terms)

An $\mathcal{L}$-structure M interprets $\mathcal{L}$-terms in the following way:

- $f(t_1, \ldots, t_n)^M = f^M(t_1^M, \ldots, t_n^M)$

$\dashv$

1.11 Definition (Extended language)

Let $\mathcal{L}$ be a first-order language and M an $\mathcal{L}$-structure. Then $\mathcal{L}(M)$ is the first-order language $\mathcal{L}$ extended with a *new* constant symbol $\bar{a}$ for each element a in $|M|$. If a is in $|M|$, then $\bar{a}$ is the *name* of a. It is required that all $\mathcal{L}(M)$-structures N interpret $\bar{a}$ as a, that is, $\bar{a}^N = a$. $\dashv$

When we evaluate $\mathcal{L}$-formulas in a structure M we will use the extended language $\mathcal{L}(M)$ and assume M to be an $\mathcal{L}(M)$-structure by interpreting $\bar{a}$ as a. This is not strictly necessary, but makes many formulations and definitions simpler. The evaluation is made precise in the following definition of what it means for a formula to be *true* in a structure.

1.12 Definition (Truth)

Let $\mathcal{L}$ be a first-order language and M an $\mathcal{L}$-structure. Assume that M is an $\mathcal{L}(M)$-structure. We define what it means for a closed $\mathcal{L}$-formula φ to be true in M, written $M \models \varphi$.

- For atomic formulas: $M \models P(t_1, \ldots, t_n)$ if $(t_1^M, \ldots, t_n^M) \in P^M$.
- $M \models \neg\varphi$ if it is *not* the case that $M \models \varphi$.
- $M \models \varphi \wedge \psi$ if $M \models \varphi$ and $M \models \psi$.
- $M \models \varphi \vee \psi$ if $M \models \varphi$ or $M \models \psi$.
- $M \models \forall x\varphi$ if $M \models \varphi[x/\bar{a}]$ for all a in $|M|$.
- $M \models \exists x\varphi$ if $M \models \varphi[x/\bar{a}]$ for an a in $|M|$.

If $M \models \varphi$, we say that M is a *model* for φ. $\dashv$

Notice that we have only defined truth for *closed* formulas, not for formulas with free variable occurrences. From now on, if not made explicit, all formulas will be closed.

1.13 Definition (Satisfiability)

An $\mathcal{L}$-formula is *satisfiable* if there is an $\mathcal{L}$-structure M in which it is true. $\dashv$

1.14 Definition (Validity)

An $\mathcal{L}$-formula φ is *valid*, written $\models \varphi$, if it is true in all $\mathcal{L}$-structures. $\dashv$

Satisfiability and validity are dual notions. If a formula is valid, then its negation is unsatisfiable. If a formula is satisfiable, then its negation is invalid.

1.15 Definition (Logical consequence)

An $\mathcal{L}$-formula φ is a *logical consequence* of a set of $\mathcal{L}$-formulas Γ, written $\Gamma \models \varphi$, if φ is true in all $\mathcal{L}$-structures that make all formulas in Γ true. $\psi \models \varphi$ abbreviates $\{\psi\} \models \varphi$. $\dashv$

Example 1.16 $\exists x Px$ is a logical consequence of $\forall x Px$ (since all models are non-empty). $P(a) \vee P(b)$ is a logical consequence of $P(a)$, but $P(a)$ is *not* a logical consequence of $P(a) \vee P(b)$.

1.3 Sequents, rules and calculi

First-order formulas and relations between these, like logical consequence, are in many settings the basic objects of study, and the object on which certain operations are made. In our case, *sequents* will be the basic objects of study.

1.17 Definition (Sequent)

A *sequent* is an object of the form $\Gamma \vdash \Delta$, where Γ and Δ are finite multisets of formulas, where a *multiset* is a set in which the multiplicity of the elements matters. Different occurrences of the same elements are distinguished. $\vdash$ is called the sequent symbol. Γ is called the *antecedent* and Δ is called the *succedent* of the sequent. If Γ' is a subset of Γ and Δ' is a subset of Δ, then $\Gamma' \vdash \Delta'$ is a *subsequent* of $\Gamma \vdash \Delta$. $\dashv$

Formula occurrences are distinguished from formulas. A formula can be represented by many different formula occurrences, e.g. in different sequents or even in the same sequent, and we will view all such formula *occurrences* as different, even though they represent the same formula.

Example 1.18 $R(a), R(b) \vdash R(c)$ is a sequent. It is *identical* to the sequent $R(b), R(a) \vdash R(c)$, but not to the sequent $R(a), R(b), R(b) \vdash R(c)$, in which the formula $R(b)$ has two occurrences in the antecedent.

Intuitively, a sequent $\Gamma \vdash \Delta$ can be seen as a judgement, to the effect that the conjunction of formulas in Γ in some sense implies the disjunction of formulas in Δ. For a sequent like $\forall x(P(x) \to Q(x)), P(a) \vdash Q(a)$, this clearly is the case. For a sequent like $P(a) \vdash Q(a)$, this is clearly *not* the case. We capture this interpretation of sequents in the notion of validity.

1.19 Definition (Validity of sequents)

A sequent $\Gamma \vdash \Delta$ is *valid* if all models that make all formulas in Γ true, also make at least one formula in Δ true. $\dashv$

Example 1.20 The sequent $\forall xPx \vdash \exists xPx$ is valid. The sequent $\exists xPx \vdash \forall xPx$ is *not* valid. The empty sequent, $\ \vdash\ $, is *not* valid.

1.21 Definition (Countermodel)

If a sequent $\Gamma \vdash \Delta$ is not valid, then there is a model that makes all formulas in Γ true and all formulas in Δ false. This is called a *countermodel* for the sequent. If a sequent has a countermodel, then it is *falsifiable*. $\dashv$

Remark. All definitions above presuppose an underlying first-order language $\mathcal{L}$. A sequent consists of $\mathcal{L}$-formulas, and validity and countermodels are thus defined with respect to $\mathcal{L}$-structures.

This is the syntax and semantics for sequents. We now turn our attention to syntactical operations on sequents, operations defined in various *calculi* by means of logical *rules*. The rest of this chapter will be devoted to the *common* parts of various calculi and rules, not the particular details of a certain calculus. Since this treatment is rather abstract, and perhaps not intuitive without prior exposure to calculi or proof systems, the reader should have a look at the various calculi introduced in later sections while reading.

The logical devices which are used to generate new sequents from old ones are called *rules*. A rule always relates one or two premisses to a conclusion. The premisses and the conclusion are in our case sequents, not formulas. As an example, if $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$ are premisses, then a rule could generate the conclusion $\Gamma \vdash \varphi \wedge \psi$. This is a *synthetical* view of rules; premisses are used to generate conclusions. The formulas, φ and ψ, are used to construct a formula $\varphi \wedge \psi$, which is part of the conclusion. The opposite view is the *analytical* view, where premisses are generated from conclusions. If the sequent $\Gamma \vdash \varphi \wedge \psi$ is our conclusion, then the same rule as above could instead have given us the two premisses. Here, $\varphi \wedge \psi$ is *taken apart* to obtain the two smaller sequents, $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$. Both the analytical and the synthetical way of applying rules are useful, but in this thesis the focus will be entirely on the analytical view. One reason for this is that it is more suitable for automated theorem proving, where one usually starts out with a complex formula or sequent in order to analyze it by examining its smaller parts.

There are two ways of defining rules. Denotationally, a rule can be defined as a set of ordered pairs, with no information whatsoever about the process of rule application or how to obtain new sequents from old ones. Operationally, a rule can be defined by providing the exact syntactical means of obtaining new sequents from old ones. With an operational definition of a rule at hand, we have the explicit machinery to generate conclusions from premisses and vice versa, in contrast to checking for membership in sets of ordered pairs. We start with a denotational definition.

1.22 Definition (Rule)

A *rule* is either (1) a relation $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$, or (2) a relation $\mathcal{R} \subseteq (\mathcal{S} \times \mathcal{S}) \times \mathcal{S}$, where $\mathcal{S}$ is the set of all sequents. In case (1), $\mathcal{R}$ relates a premiss to a conclusion, and $\mathcal{R}$ is a *one-premiss rule*. In case (2), $\mathcal{R}$ relates a pair of premisses to a conclusion and $\mathcal{R}$ is a *two-premiss rule*. Each member of $\mathcal{R}$ is called an $\mathcal{R}$-*inference*. If a sequent S is given, then the process of generating one or two appropriate sequents, such that S is the conclusion of a rule instance in which these are premisses, is a *rule application*. We require that the set $\mathcal{R}$ is decidable, i.e. that the problem of checking for membership in $\mathcal{R}$ is a decidable problem. $\dashv$

Inferences will be written like this:

$$\frac{\textsf{premiss}}{\textsf{conclusion}} \qquad\qquad \frac{\textsf{premiss 1} \qquad \textsf{premiss 2}}{\textsf{conclusion}}$$

This definition allows for many more rules than we are interested in, but the denotational view will become important later, when we want to define relations over a given set of inferences. This would be hard to do if we only had an operational definition of a rule. Actually, we are only interested in rules whose inferences have a certain form, more specifically inferences that are instances of a defining *schema*.

1.23 Definition (Schema)

A *schema* for a rule $\mathcal{R}$ is an object containing placeholders such that each $\mathcal{R}$-

inference is the result of replacing the placeholders in the schema by formula occurrences. Without loss of generality, let

$$\frac{\Gamma \vdash \psi \to \varphi, \Delta}{\Gamma, \neg\varphi \wedge \psi \vdash \Delta}$$

be an example of the schema of a rule $\mathcal{R}$. The following is then an $\mathcal{R}$-inference:

$$\frac{R(a) \vdash S(a) \to Q(a,b), \exists z P(z)}{R(a), \neg Q(a,b) \wedge S(a) \vdash \exists z P(z)}$$

This inference can be seen as the result of applying the rule $\mathcal{R}$ to the sequent $R(a), \neg Q(a,b) \wedge S(a) \vdash \exists z P(z)$.

Schemes, like inferences, have premisses and conclusions. These are the parts that become premisses and conclusions of the resulting inferences. We require that schemes satisfy the following conditions:

- Γ and Δ must occur in all premisses and conclusions (i.e. all rules of consideration are *context-sharing*) and are called *extra formulas* or *context*. They are replaced by corresponding *extra formula occurrences*.
- Formula placeholders, φ, ψ, etc., with connectives and quantifiers, can occur both in premisses and conclusions.
- A formula placeholder in a *conclusion* is called a *principal formula*, and it is replaced by a *principal formula occurrence*.
- A formula placeholder in a *premiss* is called an *active formula*, and it is replaced by an *active formula occurrence*.

In the example, $R(a)$ and $\exists z P(z)$ are the extra formula occurrences, $\neg Q(a,b) \wedge S(a)$ is the principal formula occurrence and $S(a) \to Q(a,b)$ is the only active formula occurrence of the resulting inference. $\dashv$

Schemes are very generic objects; they give rise to both denotational and operational definitions of rules. Denotationally, the set of all instances of a given schema is a rule. The schema captures the structural similarities between these inferences. Operationally, a schema can be read as a set of instructions on how to construct the appropriate premisses when a sequent is given. From now on, all rules will be given by their defining schemes.

Example 1.24 Let the following be a schema:

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \neg\neg\varphi \vdash \Delta}$$

From the sequent

$$\forall x P(x), \neg\neg P(a), \neg\neg P(b) \vdash P(a) \vee P(b)$$

the schema can be used to construct the premiss

$$\forall x P(x), P(a), \neg\neg P(b) \vdash P(a) \vee P(b)$$

Another application of the schema gives the premiss

$$\forall x P(x), P(a), P(b) \vdash P(a) \vee P(b)$$

We will presuppose some general knowledge and terminology about trees. The reader should be familiar with concepts like *root nodes*, *leaf nodes*, *branches*, etc. All trees will be displayed with the root node at the bottom. For a given branch, the nodes above a node n are called the *ancestors* of n; the nodes below n are called the *descendants* of n. The nodes directly above n are called the *immediate* ancestors of n. The one node below n is called the *immediate* descendant of n. The root node never has any descendants, and the leaf nodes never have any ancestors.

1.25 Definition (Derivation)

A *derivation* is a finitely branching tree whose nodes are sequents. Given a set of rules, the set of derivations generated by these rules is the least set that satisfies the following conditions:

- A sequent is a derivation. (This is the singleton tree consisting of one node.)
- If π is a derivation with a leaf sequent $\Gamma \vdash \Delta$, and this sequent is the conclusion of an inference with premiss $\Gamma' \vdash \Delta'$, then the tree obtained from π by adding $\Gamma' \vdash \Delta'$ above $\Gamma \vdash \Delta$ is a derivation:

$$\begin{array}{c} \Gamma \vdash \Delta \\ \vdots \\ \pi \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \dfrac{\Gamma' \vdash \Delta'}{\Gamma \vdash \Delta} \\ \vdots \\ \pi \end{array}$$

- If π is a derivation with a leaf sequent $\Gamma \vdash \Delta$, and this sequent is the conclusion of an inference with premisses $\Gamma' \vdash \Delta'$ and $\Gamma'' \vdash \Delta''$, then the tree obtained from π by adding these two sequents above $\Gamma \vdash \Delta$ is a derivation:

$$\begin{array}{c} \Gamma \vdash \Delta \\ \vdots \\ \pi \end{array} \quad \rightsquigarrow \quad \begin{array}{c} \dfrac{\Gamma' \vdash \Delta' \quad \Gamma'' \vdash \Delta''}{\Gamma \vdash \Delta} \\ \vdots \\ \pi \end{array}$$

The sequent at the root of a derivation is called a *root sequent*. ⊣

1.26 Definition (Calculus)

A *calculus* K has two components:

- A set of rules. These rules generate the K*-derivations*.
- A condition which K-derivations can meet, called the *closure condition*. If a K-derivation meets the condition, then it is called *closable*, or simply *closed*. The closure condition will typically be of the form: *for all branches in the derivation, there is a sequent with property X*, where X is the essential ingredient.

⊣

1.27 Definition (Proof)

Let K be a calculus. A *proof* of a sequent $\Gamma \vdash \Delta$ in K is a closed K-derivation in which $\Gamma \vdash \Delta$ is the root sequent. A sequent is called *provable* (in K) if it has a proof (in K). ⊣

Notice that the notions of rules, schemes, derivations and proofs are purely syntactical. We now provide the link to the semantics:

1.28 Definition (Soundness)

A calculus is *sound* if all provable sequents in the calculus are valid. ⊣

1.29 Definition (Consistent)

A calculus is *consistent* if the empty sequent $\vdash$ is unprovable. ⊣

1.30 Lemma (Consistency) A sound calculus is consistent. ⊣

Proof. Suppose the empty sequent is provable in a sound calculus. Then, it must be valid, which is impossible. □

1.31 Definition (Completeness)

A calculus is *complete* if all valid sequents are provable in the calculus. ⊣

1.4 THE CALCULUS LK

Our first calculus is called LK. This name goes back to Gentzen and his *Logische Kalküle* [20], but the calculus presented here differs from the original version in many ways: (1) In our calculus, there are no separate structural rules. Contraction is built into the quantifier rules, and weakening is not necessary for neither soundness nor completeness. (2) Our sequents consist of multisets, not *sequences*, where the order of elements also matters. (3) All our rules are *context-sharing*, while in the original LK, the rule L$\rightarrow$ is not context-sharing. (4) We don't use or have rules for the logical constant $\perp$.

Before we define the calculus, we need the notion of an axiom.

1.32 Definition (Axiom)

A sequent $\Gamma \vdash \Delta$ is an *axiom* of LK if there is an atomic formula which is in both Γ and Δ. $\dashv$

Example 1.33 $\forall x P(x) \vdash \forall x P(x)$ is *not* an axiom, since $\forall x P(x)$ is not an atomic formula. But, $P(a) \vdash P(a)$ is an axiom of LK.

1.34 Definition (The calculus LK)

The calculus LK consists of the set of rules, given by their defining schemes, in Figure 1.1. The condition for δ-inferences, that a must not occur in the conclusion, is called the *eigenparameter condition* and the term a is called an *eigenparameter*. The closure condition of LK is this: An LK-derivation is a *proof*, i.e. is closable, if all leaf sequents in the derivation are axioms. $\dashv$

The names of the rules are written to the right of each schema. The L indicates that the rule is an rule, and the R indicates that the rule is a succedent rule.

The rules are categorized into four types, following the unifying notation of Smullyan [38]: α-, β-, γ- and δ-rules. Originally, Smullyan introduced the unifying notation for semantic tableaux, but it is unproblematic to transfer it to sequent calculi: α-rules are propositional one-premiss rules, β-rules are propositional two-premiss rules, γ-rules are quantifier rules in which arbitrary, closed terms are introduced and extra copies of the quantifier formula are introduced. The δ-rules are quantifier rules in which *new* terms are introduced, terms that semantically function as *witnesses* for either the satisfiability of a formula (the L$\exists$-rule) or the unsatisfiability of a formula (the R$\forall$-rule). δ-rules have a significant role in this thesis, and will be dealt with in detail later.

α-rules

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}\ \mathsf{L}\wedge$$

$$\frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \vee \psi, \Delta}\ \mathsf{R}\vee$$

$$\frac{\Gamma, \varphi \vdash \psi, \Delta}{\Gamma \vdash \varphi \rightarrow \psi, \Delta}\ \mathsf{R}\rightarrow$$

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \neg\varphi, \Delta}\ \mathsf{R}\neg$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \neg\varphi \vdash \Delta}\ \mathsf{L}\neg$$

β-rules

$$\frac{\Gamma \vdash \varphi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \varphi \wedge \psi, \Delta}\ \mathsf{R}\wedge$$

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta}\ \mathsf{L}\vee$$

$$\frac{\Gamma \vdash \varphi, \Delta \quad \Gamma, \psi, \vdash \Delta}{\Gamma, \varphi \rightarrow \psi \vdash \Delta}\ \mathsf{L}\rightarrow$$

δ-rules

$$\frac{\Gamma \vdash \varphi[x/a], \Delta}{\Gamma \vdash \forall x\varphi, \Delta}\ \mathsf{R}\forall$$

$$\frac{\Gamma, \varphi[x/a] \vdash \Delta}{\Gamma, \exists x\varphi \vdash \Delta}\ \mathsf{L}\exists$$

γ-rules

$$\frac{\Gamma, \forall x\varphi, \varphi[x/t] \vdash \Delta}{\Gamma, \forall x\varphi \vdash \Delta}\ \mathsf{L}\forall$$

$$\frac{\Gamma \vdash \exists x\varphi, \varphi[x/t], \Delta}{\Gamma \vdash \exists x\varphi, \Delta}\ \mathsf{R}\exists$$

Figure 1.1: The rules of LK. The following conditions must hold: In γ-inferences the term t is any *closed* term. In δ-inferences the introduced term a must not occur in the conclusion.

Convention. If θ is one of these types, an occurrence of *θ-formula* is a formula occurrence which potentially can be principal in a θ-type inference. Instead of labeling derivations with L$\exists$, R$\exists$, L$\forall$, R$\forall$, the terms which are introduced, sometimes together with the inference type, will be used as labels. See the example below.

1.35 Definition (Closed branch)

If a branch of an LK-derivation contains an axiom, then the branch is *closed*. Otherwise, it is *open*. $\dashv$

An LK-proof is then equivalent to an LK-derivation in which all branches are closed

Example 1.36 (Rule dependencies) Below are proofs of $\forall x Px \vdash \forall x Px$ and $\forall x Px \vdash \exists x Px$:

$$\dfrac{\dfrac{\forall xPx, Pa \vdash Pa}{\forall xPx \vdash Pa}\,\gamma_a}{\forall xPx \vdash \forall xPx}\,\delta_a \qquad\qquad \dfrac{\dfrac{\forall xPx, Pt \vdash Pt}{\forall xPx, Pt \vdash \exists xPx}\,\gamma_t}{\forall xPx \vdash \exists xPx}\,\gamma_t$$

In the left-side proof, the order of rule application is essential. The δ-inference is below the γ-inference, which makes it possible to close the derivation with only two rule applications. If the lowermost inference were a γ-inference, then it would take at least three rule applications to close the derivation. In other words, there is a *rule dependency* between L$\forall$ and R$\forall$. In the right-side proof, the order of rule application is *not* essential. Both are γ-inferences, and there is no rule dependency between these.

Example 1.37 Here is an example where the root sequent contains a function symbol:

$$\dfrac{\dfrac{\dfrac{\forall xPxg(x), Pag(a) \vdash Pag(a)}{\forall xPxg(x), Pag(a) \vdash \exists yPay}\,\gamma_{g(a)}}{\forall xPxg(x) \vdash \exists yPay}\,\gamma_a}{\forall xPxg(x) \vdash \forall x\exists yPxy}\,\delta_a$$

1.5 SOUNDNESS OF LK

We will now prove that LK is a sound calculus; that every sequent we can prove by means of LK-rules is valid. The idea underlying this proof is the following: Since axioms are valid, and all rule applications preserve validity when going from premisses to conclusion, then the root sequent of a proof must also be valid.

A sequent is valid with respect to a first-order language $\mathcal{L}$; it must be valid in all $\mathcal{L}$-structures. To ensure that eigenparameters are interpreted in the right way, we will assume that these are *not* part of any given language $\mathcal{L}$. Instead, we will use a first-order language $\mathcal{L}^{\mathsf{par}}$, which is $\mathcal{L}$ extended with a set of new constant symbols, which are those used as eigenparameters. The root sequent of an LK-derivation will always be a closed $\mathcal{L}$-formula, while the *validity* of sequents in an LK-derivation always will be with respect to $\mathcal{L}^{\mathsf{par}}$-structures. If a sequent is valid in all $\mathcal{L}^{\mathsf{par}}$-structures, then it is also valid in all $\mathcal{L}$-structures, since the former is a superclass of the latter. The purpose of this will be clear in the proof of local soundness below, where all structures are assumed to be $\mathcal{L}^{\mathsf{par}}$-structures.

1.38 Lemma (Local soundness) (1) All axioms are valid. (2) All inferences preserve validity downwards, meaning that if the premiss, or premisses, are valid, then the conclusion is also valid. (3) Conversely, all inferences preserve falsifiability upwards, meaning that if the conclusion is falsifiable, then at least one of the premisses is falsifiable. $\dashv$

PROOF. (1) Let $\Gamma \vdash \Delta$ be an axiom. Then, there is an atomic formula which occurs in both Γ and Δ, and all models which make all formulas in Γ true must also make at least one formula in Δ true, namely the formula which is in both.

(2) L$\wedge$: Assume $\Gamma, \varphi, \psi \vdash \Delta$ is valid. We need to show that $\Gamma, \varphi \wedge \psi \vdash \Delta$ is valid. By the truth definition, Definition 1.12, this clearly is the case.

L$\vee$: Assume (1) that $\Gamma, \varphi \vdash \Delta$ and (2) that $\Gamma, \psi \vdash \Delta$ are valid. We need to show that $\Gamma, \varphi \vee \psi \vdash \Delta$ is valid. Let M be a model which makes all formulas in Γ and $\varphi \vee \psi$ true. By the truth definition, M makes either φ or ψ true. Assume without loss of generality that M makes φ true. Then, by assumption (1), M makes a formula in Δ true.

L$\neg$: Assume that $\Gamma \vdash \varphi, \Delta$ is valid and that $\Gamma, \neg\varphi$ is true in a model M. Since, M makes all formulas in Γ true, the assumption gives that either φ or a formula in Δ is true. Since M cannot make φ true (it already makes $\neg\varphi$ true), it must be the case that M makes a formula in Δ true.

R$\wedge$, R$\vee$, R$\neg$: Similar.

L$\exists$: Assume that $\Gamma, \varphi[x/a] \vdash \Delta$ is valid. We need to show that $\Gamma, \exists x\varphi \vdash \Delta$ is valid. So, let M be a model which makes all formulas in Γ and $\exists x\varphi$ true. We must show that M also makes a formula in Δ true. The assumption is not directly applicable, since we don't have that M makes $\varphi[x/a]$ true; all we have is that M makes $\exists x\varphi$ true. An extra argument is needed. By the truth definition, there must be an element b in the domain of M such that $\varphi[x/\bar{b}]$ is true in M. From the model M we construct a model M' which *is identical to* M except for the interpretation of a. We require that the interpretation of a in M' is b, that $a^{M'} = b$. The interpretations of a and $\bar{b}$ are thus the same, and since M makes $\varphi[x/\bar{b}]$ true, it must be the case that M' makes $\varphi[x/a]$ true (this is easily established by induction on the length of formulas). But then, since a does not occur in Γ or Δ, the models M and M' will interpret all formulas in these multisets in the same way. The assumption gives that there is a formula in Δ which is true in M', but this formula must also be true in M.

L$\forall$: Assume that $\Gamma, \forall x\varphi, \varphi[x/t] \vdash \Delta$ is valid. We need to show that $\Gamma, \forall x\varphi \vdash \Delta$ is valid. So, let M be a model which makes all formulas in Γ and $\forall x\varphi$ true. To get that M also makes a formula in Δ true, we need to show that M makes $\varphi[x/t]$ true; then the assumption is applicable. By the truth definition $\varphi[x/\bar{a}]$ is true in M for all elements a in the domain of M, in particular t^M. Since, t and $\overline{t^M}$ are both interpreted as t^M, $\varphi[x/t]$ must be true in M. End of proof.

R$\forall$: The proof is dual to L$\exists$. Assume that $\Gamma \vdash \varphi[x/a], \Delta$ is valid and let M be a model which makes all formulas in Γ true. The assumption gives that either $\varphi[x/a]$ is true or that there is a formula in Δ which is true. If the latter is the case, then the proof is done. If the latter is not the case, then $\varphi[x/a]$ is true and we need to show that $\forall x\varphi$ is true. Pick an arbitrary element b in the domain of M. If $\varphi[x/\bar{b}]$ is true, then we are done. This follows from the truth definition and the fact that b is arbitrary. Like in the L$\exists$-case, construct a model M' which *is identical to* M except for the interpretation of a. We require that the interpretation of a in M' is b, that $a^{M'} = b$. Then, M' must also make $\varphi[x/a]$ true. Like above, it follows (by an induction on the length of formulas) that M makes $\varphi[x/\bar{b}]$ true. End of proof.

R$\exists$: The proof is dual to L$\forall$.

(3) The converse follows easily from this. □

By repeated use of this lemma, soundness can be established. We will do this by induction on the length of proofs.

1.39 Definition (Length of a derivation)

The length of a branch β in a derivation π is the number of inferences in the branch, which is equal to the number of sequents minus one, written $|\beta|$. The *length of a derivation* π is the maximum of the lengths of all the branches in the derivation, written $|\pi|$. ⊣

1.40 Theorem (Soundness of LK)

Let π be the proof of a sequent $\Gamma \vdash \Delta$. Then $\Gamma \vdash \Delta$ is valid. ⊣

PROOF. By induction on the length of π.

Base case: $|\pi| = 0$. The only sequent in the proof is then an axiom. By Lemma 1.38 it is valid.

Inductive step: $|\pi| = n + 1$. The induction hypothesis is that all proofs of length n or less have valid root sequents. By Lemma 1.38 it suffices to show that the lowermost inference, the inference which has $\Gamma \vdash \Delta$ as conclusion, has valid premisses. (For if the premisses of this inference are valid, then by Lemma 1.38 $\Gamma \vdash \Delta$ is valid.) If it is a two-premiss inference, then both premisses are root sequents of derivations of length n or smaller. Both these derivations must be proofs, since all leaf sequents in π are axioms. By the induction hypothesis, both these proof have valid root sequents, i.e. the premisses are valid. If $\Gamma \vdash \Delta$ is the conclusion of a one-premiss inference, the proof is similar. □

1.6 Completeness of LK

We now want to prove a much stronger property; that all valid sequents are provable. Here, we must go from a universal statement, a property of all structures, to an existential statement; that there exists a proof. We will prove this by contraposition. If a sequent is not provable, then it is not valid. The idea is that if a sequent is not provable, then, by systematically using the rules of LK, it is possible to construct a model which falsifies the sequent.

1.41 Definition (Herbrand universe)

The *Herbrand universe* of a set of terms T, written $H(T)$, is the set of *closed* terms which can be generated from T, together with a dummy constant, in case there are no constants in T. More precisely, let $H(T)$ be the least set that satisfies the following conditions:

- $H(T)$ contains all closed terms of T. If there are no closed terms in T, then a dummy constant t is added to $H(T)$.
- If f is an n-ary function symbol in T and $t_1, \ldots, t_n$ are terms in $H(T)$, then $f(t_1, \ldots, t_n)$ is in $H(T)$.

The Herbrand universe of a set of sequents is the Herbrand universe of the terms occurring in the sequents. The Herbrand universe of a branch in a derivation is the Herbrand universe of all the sequents in the branch. ⊣

Remark. The specification of how to *systematically* apply LK-rules in order to get a proof, if there is one, is defined with a fair strategy. This can be done in several ways. Since the main focus here is not completeness, there will be minor simplifications. Like in Waaler [44] we will allow the existence of infinite derivation trees, limit objects, generated by fair strategies. Infinite derivation trees will have inferences just like finite derivation trees, but infinitely many of them. A branch in an infinite derivation tree is open if there is no sequent in the branch which is an axiom. Also, such limit objects will only be used in this context. Another approach is to define *sequences* of derivations trees, and define fairness in terms of such sequences. We will not do so here.

1.42 Definition (Fair strategy for LK)

A *strategy* for LK is a specification of how to apply rules to non-axiomatic sequents for all derivations in LK. A strategy is *fair* if every limit derivation π, possibly infinite, generated by this strategy has the following properties: (1) If φ is an α-, β- or δ-formula which occurs in a branch b which is not closed, then φ is principal in somewhere in b. (2) If φ is a γ-formula of the form $Qx\psi$ which occurs in a branch b, then for all terms t in the Herbrand universe of b, the formula $\psi[x/t]$ is an active formula occurrence somewhere in b. Essentially, γ-formulas must be instantiated with all closed terms occurring in, or generated from, b. ⊣

The purpose of a fair strategy is this: If a root sequent is provable, then by following a fair strategy we should eventually reach a proof. Otherwise, a fair strategy should provide enough information for the construction of a countermodel for the sequent.

Example 1.43 A fair strategy for the sequent $\forall x \exists y Pxy \vdash$ generates a single infinite branch. We shall soon see that this infinite branch gives rise to a countermodel for the root sequent. Each δ-inference has been labeled with its type and the eigenparameter that it introduces.

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\vdots}{\forall x \exists y Pxy, Pa_1a_2, Pa_0a_1, Pta_0 \vdash}}{\forall x \exists y Pxy, \exists y Pa_1y, Pa_0a_1, Pta_0 \vdash}\,\delta_{a_2}}{\forall x \exists y Pxy, Pa_0a_1, Pta_0 \vdash}\,\gamma_{a_1}}{\forall x \exists y Pxy, \exists y Pa_0y, Pta_0 \vdash}\,\delta_{a_1}}{\cfrac{\forall x \exists y Pxy, Pta_0 \vdash}{\cfrac{\forall x \exists y Pxy, \exists y Pty \vdash}{\forall x \exists y Pxy \vdash}\,\gamma_t}\,\delta_{a_0}}\,\gamma_{a_0}
$$

Example 1.44 The condition that γ-formulas must be instantiated with all terms *generated* from the branch is necessary. Without this condition, a strategy could result in no rule applications at all for the unprovable sequent $\forall x P(x, fx) \vdash$. A fair strategy, on the other hand, would generate:

$$
\cfrac{\cfrac{\cfrac{\vdots}{\forall x P(x, fx), P(ft, fft), P(t, ft) \vdash}}{\forall x P(x, fx), P(t, ft) \vdash}\,\gamma_{ft}}{\forall x P(x, fx) \vdash}\,\gamma_t
$$

Example 1.45 Condition (1) is also necessary. The sequent

$$\forall x P f(x), Qa \wedge Qa \vdash Qa$$

is obviously provable. By solely applying $\mathsf{L}\forall$, which is a non-fair strategy, a proof is ever obtained.

1.46 Theorem (Model existence)

Let $\Gamma \vdash \Delta$ be a non-provable sequent and π be the object obtained from a fair strategy for $\Gamma \vdash \Delta$. Then, $\Gamma \vdash \Delta$ has a countermodel. $\dashv$

Proof. Note that π can have infinite branches, and will have so if there are γ-inferences in an open branch with an infinite Herbrand universe. While γ-inferences introduce new formulas, all the other rules are strictly analytical in the sense that they only generate smaller objects and don't introduce new ones. There can only be countably many inferences in any given branch, since there are only countably many terms. Since π is not a proof, there must be at least one open branch β in π. If π has finitely many inferences, this is obvious. If π is infinite, the existence of an open branch follows from Königs Lemma[1]. Let β^+ consist of all formula occurrences in any antecedent of β and β^- consist of all formula occurrences in any succedent of β.

Let $\mathcal{L}$ be the language containing all function and predicate symbols occurring in β. We now define an $\mathcal{L}$-structure M:

- The domain of M is the set of all terms occurring in β
- Any $\mathcal{L}$-term is interpreted as itself. Formally: $f^M(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$
- For any atomic $\mathcal{L}$-formula $P(t_1, \ldots, t_n)$, $M \models P(t_1, \ldots, t_n)$ holds if and only if $P(t_1, \ldots, t_n)$ is in β^+.

Claim: M makes all formulas in β^+ true and all formulas in β^- false simultaneously.

Proof (of claim). By induction on the construction of formulas in β. The base case follows by definition of M. The inductive step goes like this:

Assume that φ is of the form $\psi_1 \wedge \psi_2$: If φ is in β^+, then by fairness ψ_1 and ψ_2 are also in β^+. By the induction hypothesis, ψ_1 and ψ_2 are true in M. By the truth definition, φ is also true in M. If φ is in β^-, then by fairness either ψ_1 or ψ_2 is also in β^-. In either case, by the induction hypothesis and the the truth definition, φ is false in M.

If φ is of the form $\psi_1 \vee \psi_2$ or $\psi_1 \to \psi_2$, then the proof is exactly dual to this.

[1]See e.g. Fitting [19] for details.

Assume that φ is of the form $\neg\psi$: If φ is in β^+, then by fairness ψ is in β^-. By the induction hypothesis, ψ is false in M. By the truth definition φ is true in M. If φ is in β^-, then the proof is exactly dual to this.

Assume that φ is of the form $\exists x\psi$: If φ is in β^+, then by fairness, there is a formula $\psi[x/a]$ in $\beta^+$for some term a. By the induction hypothesis, $\psi[x/a]$ is true in M. By the truth definition φ is also true in M. If φ is in β^-, then by fairness, $\psi[x/t]$ is in β^+ *for all terms t in the domain of M*. Thus, by the truth definition, φ is true in M. (End of proof of claim.) □

Since all formulas in Γ are in β^+ and all formulas in Δ are in β^-, M is a countermodel for $\Gamma \vdash \Delta$. □

1.47 Theorem (Completeness of LK)

Let $\Gamma \vdash \Delta$ be a valid sequent. If $\Gamma \vdash \Delta$ is not provable, then by the model existence theorem, $\Gamma \vdash \Delta$ has a countermodel, in which case it is not valid. So, $\Gamma \vdash \Delta$ must be provable. $\dashv$

Chapter 2

Free variable sequent calculi

Since we are mainly interested in calculi for the purpose of proof search procedures, keeping in mind possible *implementations* of the calculi we introduce, LK is in many ways not the ideal calculus. One crucial difficulty lies in implementing the γ-rules. In order to obtain a proof of a sequent, it may be necessary to introduce many terms. But, which terms should be chosen? And when should we stop introducing new terms? After all, the γ-rules allow us to introduce *any* terms whatsoever.

Example 2.1 The sequent $\forall x P(x) \vdash P(a) \wedge P(b) \wedge P(c) \wedge P(d) \wedge P(e)$ is provable. It requires five applications of the γ-rule $\mathsf{L}\forall$; one for each branch in the tree.

Example 2.2 The sequent $\forall x P(x) \vdash P(a) \vee P(b) \vee P(c) \vee P(d) \vee P(e)$ is also provable. But, in contrast, it requires only *one* application of the γ-rule. A proof search may nevertheless result in five applications before a proof is reached.

A naive way of choosing terms for the γ-inferences can be found in the definition of a fair strategy, where it is required that γ-formulas are instantiated with all terms occurring in, or generated from, the branches in which they occur. It is naive, since no intelligent selection of terms takes place at all. When proving completeness, we were only interested in obtaining *enough* terms as to guarantee the construction of a countermodel. Of course, a fair strategy always ends up with a proof if there is one; this is the content of a strategy being fair. The problem is that this proof may be unnecessary long, and that it takes unnecessary long time to find it.

A less naive way of choosing terms can still lead to many instantiations of γ-formulas before a proof is found; the terms that are necessary for the branches to close, and which eventually lead to a proof, might not be known at the time of instantiation.

A solution to this is to let γ-rules introduce *free variables* instead of arbitrary terms. Then, the actual value of a term can be delayed until more information is gathered, and unnecessary applications of γ-rules can be avoided. The idea is to postpone these decisions as long as possible until all branches can be closed by means of a closing unifier, a substitution which sends free variables to terms. The search for a proof is then done almost entirely at the level of terms, not at the level of derivations. But, the introduction of free variables for γ-rules poses a natural question for the δ-rules: what terms should δ-rules introduce? In LK, the eigenparameter condition heavily restrains which terms the δ-rules can introduce; a term must be new relative to the conclusion and function as a witness of either satisfiability or unsatisfiability. With free variables, how can we ensure that the terms which δ-rules introduce are *fresh* in the same way? There are two approaches to this question. First, we *can* ensure that the terms are fresh by extending the notion of a parameter in the following way. If $u_1, \ldots, u_n$ are all the variables occurring in a branch, then a δ-rule could introduce a term $f(u_1, \ldots, u_n)$, for a new function symbol f, called a Skolem function. In this way, no matter what $u_1, \ldots, u_n$ are instantiated with, the term $f(u_1, \ldots, u_n)$ is instantiated with a different term. Second, it is not necessary that the terms introduced by δ-rules are fresh. The eigenparameter condition is actually too strong; there are many ways of systematically introducing and reusing terms, thus *liberalizing* the δ-rules, without affecting neither soundness nor completeness of a calculus.

Below is a summary of the approaches that exist.

- δ-rule. This is the original δ-rule introduced by Fitting [19] in 1990, and it corresponds exactly to the description above. The arguments of the Skolem function are all the variables occurring in the branch. The second edition of his book, from 1996, used a δ^+-rule.
- δ^+-rule. Introduced by Hähnle and Schmitt [28] in 1991. The arguments of the Skolem function are exactly the free variables occurring in the δ-formula at hand.
- δ^{++}-rule. Introduced by Beckert, Hähnle and Schmitt [10] in 1993. It is like the δ^+-rule, but the number of different Skolem functions that must be used is reduced. When the δ^{++}-rule is applied to two formulas which are identical up to renaming of free and bound variables, then the same Skolem function is introduced.
- δ^*-rule. Introduced by Baaz and Fermüller [4] in 1995. They define a set of *relevant* free variables occurring in a δ-formula, which is a subset of the free variables occurring in it. When the δ^*-rule is applied, the arguments of the Skolem function are these variables only.

- δ^{*^*}-rule. Introduced by Cantone and Asmundo [16] in 1998. It is based on the combination of (a recursive generalization of) the concept of relevant variables, together with a notion of *key formulas*. A similar reuse of Skolem functions as in the δ^{++}-rule is utilized.

- δ^{ϵ}-rule. Introduced by Giese and Ahrendt [25] in 1999; previously in Giese's Diploma Thesis [21] from 1998. This rule is based on Hilbert's ϵ-calculus [29], and instead of Skolem functions, ϵ*-terms* are introduced. (ϵ-terms are objects which function as witnesses by virtue of their syntactical structure.)

Historically, the use of free variables in first-order calculi can be traced back to Prawitz [36] and Kanger [32], who called them "dummies".

We will now introduce some terminology common for all calculi with free variables. First of all, we separate the terms which are introduced by γ- and δ-rules from the terms which occur in a given first-order language. From now on, we will refer to the variables which are part of every first-order language, and which are bound by quantifiers, as *quantification variables*. Instead of using these as the terms which γ-rules introduce, we use a different set of variables, called *instantiation variables*. Likewise, the δ-rules will now introduce *Skolem functions* with instantiation variables as arguments. Together with the function symbols of a first-order language, instantiation variables and Skolem functions generate the set of *instantiation terms*.

2.3 Definition

Let $\mathcal{L}$ be a first-order language. Let $\mathcal{U}$ be an infinite and countable set of *instantiation variables*, and let $\mathcal{S}$ be an infinite and countable set of *Skolem functions*. These should be disjoint sets of symbols and not contain any symbols from $\mathcal{L}$.

The set of *instantiation terms* for $\mathcal{L}$ is the least set such that:

- All instantiation variables are instantiation terms.

- If f is a Skolem function or a function symbol in $\mathcal{L}$ of arity n and $t_1, \ldots, t_k$ are instantiation terms, then $f(t_1, \ldots, t_k)$ is an instantiation term. If f is a Skolem function, then $f(t_1, \ldots, t_k)$ is called a *Skolem term*.

From now on, all first-order languages (and related notions) will be considered to be contain instantiation terms. $\mathcal{L}$-formulas *with instantiation terms* are defined exactly like $\mathcal{L}$-formulas in Definition 1.3, except for the base case:

- If P is an n-ary predicate symbol of $\mathcal{L}$ and $t_1, \ldots, t_n$ are $\mathcal{L}$-terms *or instantiation terms*, then $P(t_1, \ldots, t_n)$ is an $\mathcal{L}$-formula with instantiation terms.

Substitutions for such first-order languages are defined like before, except that quantification variables can be sent to instantiation terms as well as terms of the language. A *ground instantiation term* is defined to be an instantiation term without any occurrences of instantiation variables in it. ⊣

In a free variable calculus all formulas, except for the formulas in the root sequent, will be taken from a first-order language with instantiation terms. Also, such formulas will be generated by the rules of a calculus and will not exist outside such a context.

Remark. Instantiation terms do not contain quantification variables. Therefore, no variables can become bound as a result of replacing variables by instantiation terms.

All substitutions to this point have been functions with quantification variables in the domain; instantiation terms enable us to speak of, and give a separate treatment to, substitutions with instantiation variables in the domain.

2.4 Definition (Substitution for instantiation terms)

A *substitution for instantiation terms* is a function σ from instantiation variables to instantiation terms. $\varphi\sigma$ denotes the formula φ where each instantiation variable u in the domain of σ has been replaced by $u\sigma$. $\Gamma \vdash \Delta$ denotes the sequent $\Gamma \vdash \Delta$ where each instantiation variable u in the domain of σ has been replaced by $u\sigma$. σ is *ground* if $u\sigma$ is a ground instantiation term for all u in the domain of σ. ⊣

2.5 Definition (Unifier)

If s and t are instantiation terms, σ is a substitution for instantiation terms and $s\sigma = t\sigma$, then σ is a *unifier* of s and t. If φ and ψ are two formulas with instantiation terms and $\varphi\sigma = \psi\sigma$, then σ is a *unifier* of φ and ψ. A unifier is also called a *closing substitution*. ⊣

2.6 Definition (Skeleton)

A *skeleton* is a derivation in which there are instantiation variables. ⊣

2.1 Proof transformations

A free variable calculus which behaves exactly like the *ground* calculus LK will soon be introduced, and it will match the informal description given in the beginning of this chapter; the Skolemization will be done with respect to all instantiation variables occurring in the conclusion of a sequent. The point of introducing this calculus is not that it is particularly interesting *in itself*, but rather that it gives us a firm basis to which we can relate other calculi with different, and more liberal, quantifier rules. This calculus will be viewed as a canonical example of a calculus which only generates “LK-like” proofs. (See Definition 4.30.) Soundness and completeness of this calculus are established analogous to LK.

The idea is that if we have constructed a free variable calculus X and we are able to effectively *transform* any X-proof of a sequent into an LK^{δ}-proof of the same sequent, then we have established soundness of this calculus. Then, any X-provable sequent is LK^{δ}-provable, and therefore valid. Furthermore, if we want to show completeness, all we need to do is to show how any LK^{δ}-proof of a sequent effectively can be *transformed* into an X-proof of the same sequent. Since all valid sequents are LK^{δ}-provable, by completeness of LK^{δ}, we obtain X-proofs for all valid sequents.

Such soundness proofs are syntactical in nature; by syntactical operations on derivation trees, we construct new derivation trees and show properties of these. In the authors view, such soundness proofs are more *constructive* than soundness proofs which are semantical in nature. From a syntactical soundness proof it is often possible to extract more information than just the soundness the calculus. Imagine some obscure system with highly non-intuitive rules and that a semantical soundness proof for this calculus has been provided, possibly with equally obscure techniques. We ask the system: “Does $\mathbf{P} = \mathbf{NP}$?”; it answers “yes” and provides a – let us assume, rather obscure – proof. How are we then to interpret the given proof? If the rules are highly non-intuitive, the proof might altogether be impossible to understand. We would be better off if we had an effective way of translating the given proof into a proof in a system which had more understandable rules. This is exactly what a syntactical soundness proof can provide!

In automated reasoning, it is crucial to minimize the search space and eliminate non-determinism whenever possible, mainly due to space and time limitations of computers. One way of doing this is to postpone choice points as long as possible and examine many different parts of a search space simultaneously until a proof can be found, which is what we do in free variable calculi. This is a technique which is typically *not* used by human reasoners, which, in contrast, most often only reason from a small set of *assumptions* at any given time. The search space is explored, but in a very different way; often human reasoners are able to exclude large parts of a given search space, which automated theorem provers are forced to search exhaustively. Interactive theorem proving is the field of theorem proving which incorporates human interaction; for example, by allowing users to introduce *lemmas* or *cuts* into a proof.

The two paradigms of *search-oriented calculi* on the one side and *interaction-oriented calculi* on the other, are described excellently in [40]. The syntactical soundness proofs provided in this thesis correspond exactly to their *proof transformations* between search-oriented and interaction-oriented calculi.

2.2 The calculus LK^δ

The whole point of LK^δ is to have a free variable calculus which behaves exactly like LK. This will greatly facilitate soundness proofs of other free variable calculi, since in many cases where a proof is given, we can translate this proof to a proof in LK^δ.

2.7 Definition (σ-axiom)

A sequent $\Gamma, \varphi \vdash \psi, \Delta$ is a σ*-axiom* if φ and ψ are atomic formulas and σ is a unifier of φ and ψ. $\dashv$

2.8 Definition (The calculus LK^δ)

The calculus LK^δ is obtained by changing the quantifier rules and the closure condition of LK:

δ-rules	γ-rules
$$\frac{\Gamma \vdash \varphi[x/f(\vec{u})], \Delta}{\Gamma \vdash \forall x \varphi, \Delta}\ \mathsf{R}\forall$$	$$\frac{\Gamma, \forall x \varphi, \varphi[x/u] \vdash \Delta}{\Gamma, \forall x \varphi \vdash \Delta}\ \mathsf{L}\forall$$
$$\frac{\Gamma, \varphi[x/f(\vec{u})] \vdash \Delta}{\Gamma, \exists x \varphi \vdash \Delta}\ \mathsf{L}\exists$$	$$\frac{\Gamma \vdash \exists x \varphi, \varphi[x/u], \Delta}{\Gamma \vdash \exists x \varphi, \Delta}\ \mathsf{R}\exists$$

Figure 2.1: The rules of LK^δ. In all γ-inferences, u must be an instantiation variable which does not already occur in the *derivation* when it is introduced. In all δ-inferences, $\vec{u}$ is a sequence of all instantiation variables occurring in the conclusion, and f must be a Skolem function which does not occur in the *conclusion*.

The closure condition for LK^δ is this: An LK^δ-skeleton π is closable if there is a substitution σ such that all leaf sequents in π are σ-axioms. The proof will be denoted $\langle \pi, \sigma \rangle$. $\dashv$

Convention. We will use u, v, w, etc. as meta-symbols for instantiation variables and f, g, h, etc. as meta-symbols for function symbols. Usually, a, b, c, etc. will be used as meta-symbols for constant symbols. It will always be clear from the context, usually the derivations, whether a symbol f stands for a Skolem function or a function symbol from a first-order language.

Example 2.9 (Rule dependencies) This example is analogous to Example 1.36. Below are proofs of $\forall x Px \vdash \forall x Px$ and $\forall x Px \vdash \exists x Px$ in LK^δ:

$$\dfrac{\dfrac{\dfrac{u \mapsto a}{\forall x Px, Pu \vdash Pa}}{\forall x Px \vdash Pa}\,\gamma_u}{\forall x Px \vdash \forall x Px}\,\delta_a \qquad \dfrac{\dfrac{\dfrac{u_1 \mapsto u_2}{\forall x Px, Pu_1 \vdash Pu_2, \exists x Px}}{\forall x Px, Pu_1 \vdash \exists x Px}\,\gamma_{u_2}}{\forall x Px \vdash \exists x Px}\,\gamma_{u_1}$$

In the right-side proof, the order of the rule application is *not* essential. Both are γ-inferences, and there is no rule dependency between these.

In the left-side proof, the order of the rule applications *is* essential. The δ-inference is below the γ-inference, which makes it possible to close the derivation with only two rule applications. If the lowermost inference were a γ-inference, then it would take at least three rule applications to close the derivation. In other words, there is a *rule dependency* between $\mathsf{L}\forall$ and $\mathsf{R}\forall$ in LK^δ, just like in LK. The skeleton below shows that three rule applications is necessary.

$$\dfrac{\dfrac{\dfrac{\dfrac{v \mapsto fu}{\forall x Px, Pv, Pu \vdash Pfu}}{\forall x Px, Pu \vdash Pfu}\,\gamma_v}{\forall x Px, Pu \vdash \forall x Px}\,\delta_{fu}}{\forall x Px \vdash \forall x Px}\,\gamma_u$$

After two rule applications, the sequent $\forall x Px, Pu \vdash Pfu$ has no closing substitution. (If u/fu is applied to the sequent, $\forall x Px, Pfu \vdash Pffu$ is obtained.)

Example 2.10 Here is an example, analogous to Example 1.37, where the root sequent contains a function symbol g.

$$\dfrac{\dfrac{\dfrac{\dfrac{u \mapsto a, v \mapsto g(a)}{\forall x Pxg(x), Pug(u) \vdash Pav, \exists y Pay}}{\forall x Pxg(x), Pug(u) \vdash \exists y Pay}\,\gamma_v}{\forall x Pxg(x) \vdash \exists y Pay}\,\gamma_u}{\forall x Pxg(x) \vdash \forall x \exists y Pxy}\,\delta_a$$

The following lemma will be used for all free variable calculi.

2.11 Lemma (Grounding) Let $\langle \pi, \sigma \rangle$ be a proof. Then, there is a *ground* substitution σ' such that $\langle \pi, \sigma' \rangle$ is a proof of the same sequent. $\dashv$

PROOF. Let π' be the skeleton $\pi\sigma$ where σ has been applied to all formulas in the skeleton and assume that σ is not ground. Let $u_1, \ldots, u_n$ be all the instantiation variables occurring in π' and let a be a constant which does not occur in the skeleton. Let τ be a substitution (for instantiation terms) which sends any of these instantiation variables to a, that is $\tau(u_i) = a$ for $1 \leq i \leq n$. Observe that τ closes π'. Then, $\sigma\tau$, the composition of σ and τ, is a substitution which is ground such that $\langle \pi, \sigma\tau \rangle$ is a proof. $\square$

2.3 SOUNDNESS AND COMPLETENESS OF LK^δ

We will now prove that LK^δ is a sound calculus; that every sequent that we can prove by means of LK^δ-rules is valid. The idea is not difficult. Since the δ-rules Skolemize with respect to all instantiation variables occurring in the conclusion, each inference essentially satisfies the eigenparameter condition of LK, so by replacing all occurrences of instantiation variables in a skeleton with *ground* terms a local soundness lemma can be shown.

2.12 Lemma (Local soundness) Let $\langle \pi, \sigma \rangle$ be a proof where σ is a ground substitution. Let $\pi\sigma$ be the object obtained by applying σ to all instantiation terms occurring in π. (1) All axioms in $\pi\sigma$ are valid. (2) All inferences in $\pi\sigma$ preserve validity downwards, meaning that if the premiss, or premisses, are valid, then the conclusion is also valid. $\dashv$

PROOF. Analogous to the proof of local soundness for LK, Lemma 1.38. $\square$

2.13 Theorem (Soundness of LK^δ)

Let $\langle \pi, \sigma \rangle$ be a proof in LK^δ of a sequent $\Gamma \vdash \Delta$. Then $\Gamma \vdash \Delta$ is valid. $\dashv$

PROOF. By Lemma 2.11 there is a ground substitution σ' such that $\langle \pi, \sigma' \rangle$ is a proof of $\Gamma \vdash \Delta$. The proof is by induction on the length of $\pi\sigma'$ and proceeds by repeated applications of Lemma 2.12, analogous to the the Soundness theorem (Theorem 2.13) for LK. $\square$

2.14 Theorem (Completeness of LK^δ)

All valid sequents are LK^δ-provable. $\dashv$

PROOF. Let $\Gamma \vdash \Delta$ be a valid sequent. By completeness of LK, there is an LK-proof π of $\Gamma \vdash \Delta$. By simulating this derivation in LK^δ, a skeleton π'

and a closing substitution σ are obtained. See Example 2.15 below for an indication of how this can be done. □

Example 2.15 We show how an LK-proof can be translated into an LK^{δ}-proof. (Some unused copies of γ-formulas are not displayed.) An LK-proof for the sequent $\forall x\exists yPxy \vdash \forall x\exists y(Pxy \wedge \exists zPyz)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x\exists yPxy, Pab \vdash Pab \qquad \dfrac{\dfrac{\dfrac{\dfrac{Pbc, Pab \vdash Pb\boxed{c}}{Pb\boxed{c}, Pab \vdash \exists zPbz}\,\gamma_c}{\exists yP\boxed{b}y, Pab \vdash \exists zPbz}\,\delta_c}{\forall x\exists yPxy, Pab \vdash \exists zPbz}\,\gamma_b}{\forall x\exists yPxy, Pab \vdash Pa\boxed{b} \wedge \exists zP\boxed{b}z}}{\forall x\exists yPxy, Pa\boxed{b} \vdash \exists y(Pay \wedge \exists zPyz)}\,\gamma_b}{\forall x\exists yPxy, \exists yP\boxed{a}y \vdash \exists y(Pay \wedge \exists zPyz)}\,\delta_b}{\forall x\exists yPxy \vdash \exists y(P\boxed{a}y \wedge \exists zPyz)}\,\gamma_a}{\forall x\exists yPxy \vdash \forall x\exists y(Pxy \wedge \exists zPyz)}\,\delta_a$$

The LK^{δ}-skeleton obtained by applying the corresponding rules in the same order:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x\exists yPxy, Puf(u) \vdash Pav \qquad \dfrac{\dfrac{\dfrac{\dfrac{Pwg(u,v,w), Puf(u) \vdash Pv\boxed{w'}}{Pw\boxed{g(u,v,w)}, Puf(u) \vdash \exists zPvz}\,\gamma_{w'}}{\exists yP\boxed{w}y, Pufu \vdash \exists zPvz}\,\delta_{g(u,v,w)}}{\forall x\exists yPxy, Pufu \vdash \exists zPvz}\,\gamma_w}{\forall x\exists yPxy, Puf(u) \vdash Pa\boxed{v} \wedge \exists zP\boxed{v}z}}{\forall x\exists yPxy, Pu\boxed{f(u)} \vdash \exists y(Pay \wedge \exists zPyz)}\,\gamma_v}{\forall x\exists yPxy, \exists yP\boxed{u}y \vdash \exists y(Pay \wedge \exists zPyz)}\,\delta_{f(u)}}{\forall x\exists yPxy \vdash \exists y(P\boxed{a}y \wedge \exists zPyz)}\,\gamma_u}{\forall x\exists yPxy \vdash \forall x\exists y(Pxy \wedge \exists zPyz)}\,\delta_a$$

The terms are boxed at the places where they are introduced. By comparing which terms are introduced where, a closing substitution for the skeleton is obtained. A table of correlations is given below.

	γ-terms				δ-terms		
LK	a	b	b	c	a	b	c
LK^{δ}	u	v	w	w'	a	$f(u)$	$g(u,v,w)$

From this table the resulting σ is constructed:
$\{u/a, v/fa, w/f(a), w'/g(a, fa, fa)\}$.

2.4 Formula trees

Observe that formulas have an implicit tree structure; the atomic formulas are like leaf nodes, binary connectives are like nodes with two ancestor nodes, etc. Before diagram representations are introduced, this tree structure is made explicit by the notion of a *formula tree*. We still want to distinguish formulas from formula occurrences, and formula trees make this possible in an elegant way. Formula trees will be the basic objects of which formula *occurrences* are representations, i.e. each formula occurrence will have an underlying formula tree. Two different occurrences of a formula will have two different underlying formula trees, but these formula trees will in return only differ with respect to a *copy number* (defined below). In this way, an implementation would not need to represent each formula occurrence by its own formula tree object, but rather by a single object together with a natural number.

2.16 Definition (Formula tree)

A *formula tree* is a representation of an occurrence of a formula φ as a tree. Each node in the tree represents an occurrence of a subformula ψ and has the following components:

- A *label*, which is the major connective of ψ (possibly with quantification variables), or ψ if ψ is atomic.
- An *index pair* u_m^n, which is a unique identifier of the position in the tree; the subscript m is a natural number called the *occurrence number* of the node and the superscript n is a natural number called the *copy number* of the node. All copy numbers in a formula tree should be identical and all occurrence numbers should be distinct.
- A *polarity*, which is **L** or **R** . The polarity indicates whether a formula occurs in the antecedent (left side) or the succedent (right side) of a sequent. Formulas with polarity **L** correspond to formulas in the antecedent of a sequent and vice versa. The polarity of a node is determined by the polarity and major connective of its descendant node.
- A *principal type*, which is α, β, γ, δ, or empty if ψ is atomic. The principal type of a node is determined by the polarity and the label of that node. The motivation for having a principal type θ for a node is that the formula occurrence corresponding to this node is potentially principal in a θ-inference. A node with principal type θ will be referred to as a θ-node.

We will refer to a node by either giving its index pair or the subformula it represents.

Here is how the principal type and the polarity is determined. The descendant label is in each case written *below* the ancestor labels; in the same way that descendant nodes are written below ancestor nodes in the formula tree.

polarity:	$\varphi\ \mathbf{R}, \psi\ \mathbf{R}$	$\varphi\ \mathbf{L}, \psi\ \mathbf{L}$	$\varphi\ \mathbf{R}, \psi\ \mathbf{L}$	$\varphi\ \mathbf{L}$	$\varphi\ \mathbf{R}$
princ. type α:	$(\varphi \land \psi)\ \mathbf{R}$	$(\varphi \lor \psi)\ \mathbf{L}$	$(\varphi \to \psi)\ \mathbf{L}$	$(\neg\varphi)\ \mathbf{R}$	$(\neg\varphi)\ \mathbf{L}$
polarity:	$\varphi\ \mathbf{L}, \psi\ \mathbf{L}$	$\varphi\ \mathbf{R}, \psi\ \mathbf{R}$	$\varphi\ \mathbf{L}, \psi\ \mathbf{R}$		
princ. type β:	$(\varphi \land \psi)\ \mathbf{L}$	$(\varphi \lor \psi)\ \mathbf{R}$	$(\varphi \to \psi)\ \mathbf{R}$		
polarity:	$\varphi\ \mathbf{R}$	$\varphi\ \mathbf{L}$			
princ. type γ:	$(\forall x\varphi)\ \mathbf{R}$	$(\exists x\varphi)\ \mathbf{L}$			
polarity:	$\varphi\ \mathbf{L}$	$\varphi\ \mathbf{R}$			
princ. type δ:	$(\forall x\varphi)\ \mathbf{L}$	$(\exists x\varphi)\ \mathbf{R}$			

Formula trees enable us to be somewhat more precise, and we will assume that the following is the case:

- All formula occurrences have a unique formula tree associated with it, thereby a unique index pair. When it is called for, we will write the formula occurrence together with its index pair. φ^n_m stands for a formula occurrence whose formula tree has ^n_m as its root node
- A formula occurrence to which a substitution has been applied is associated with a formula tree *together* with the substitution. The substitution is not encoded into the formula tree.
- In a root sequent of a skeleton, the formula trees underlying the formula occurrences will satisfy these conditions: (1) No two occurrence numbers are identical. (2) All copy numbers equal 1.

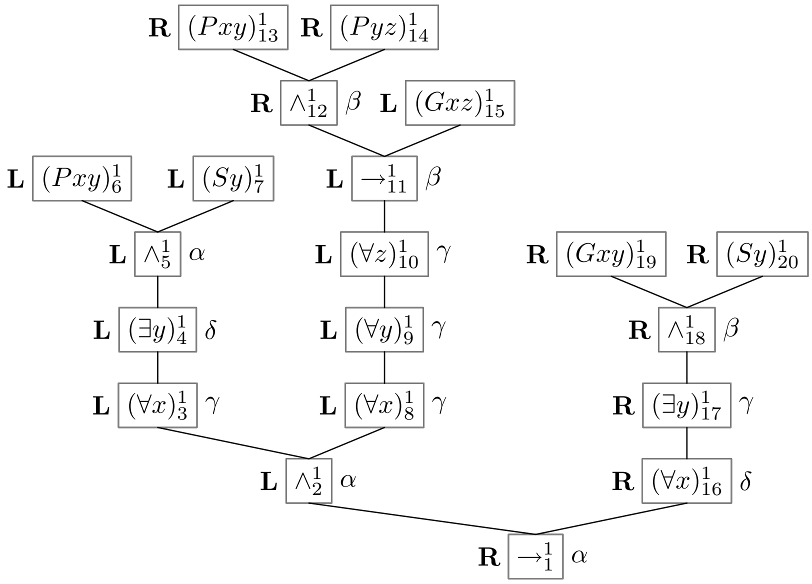
$\dashv$

Remark. The last assumption above easily gives that *any* sequent of a skeleton contains only distinct index pairs. By the assumption, all index pairs of a root sequent are distinct. Furthermore, all rules preserve this property.

Example 2.17 Below is a formula tree for the formula

$$(\forall x\exists y(Pxy \land Sy) \land \forall x\forall y\forall z((Pxy \land Pyz) \to Gxz)) \to \forall x\exists y(Gxy \land Sy)$$

The polarity is displayed to the left of each node, the principal type is displayed to the right and all labels have an index pair attached to them.



If we want to give this formula with its index pairs, we sometimes use the following notation:

$$((\underset{3}{\forall x}\underset{4}{\exists y}(\underset{6}{Pxy} \underset{5}{\wedge} \underset{7}{Sy}) \underset{2}{\wedge} \underset{8}{\forall x}\underset{9}{\forall y}\underset{10}{\forall z}((\underset{13}{Pxy} \underset{12}{\wedge} \underset{14}{Pyz}) \underset{11}{\rightarrow} \underset{15}{Gxz})) \underset{1}{\rightarrow} \underset{16}{\forall x}\underset{17}{\exists y}(\underset{19}{Gxy} \underset{18}{\wedge} \underset{20}{Sy}))^1$$

This is much more readable than the formula with all parentheses and index pairs shown:

$$\begin{gathered}(((\forall x(\exists y((Pxy)^1_6 \wedge (Sy)^1_7)^1_5)^1_4)^1_3 \\ \wedge(\forall x(\forall y(\forall z(((Pxy)^1_{13} \wedge (Pyz)^1_{14})^1_{12} \rightarrow (Gxz)^1_{15})^1_{11})^1_{10})^1_9)^1_8)^1_2 \\ \rightarrow (\forall x(\exists y((Gxy)^1_{19} \wedge (Sy)^1_{20})^1_{18})^1_{17})^1_{16})^1_1\end{gathered}$$

Example 2.18 If $\varphi \rightarrow \psi$ has polarity 1, then φ will have polarity 0 and ψ will have polarity 1. Here are two intuitions for this: First, if the formula $\varphi \rightarrow \psi$ is true, then there are two possibilities. Either, φ is false, which corresponds to polarity 0, or ψ is true, which corresponds to polarity 1.

Second, that $\varphi \rightarrow \psi$ has polarity 1, indicates that it occurs in the antecedent of a sequent. If it is principal in an inference, the situation is this:

$$\frac{\Gamma, \vdash \Delta, \varphi \quad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \rightarrow \psi \vdash \Delta}$$

φ occurs in the succedent and is associated with polarity 0; φ occurs in the antecedent and is associated with polarity 1.

2.19 Definition (Subformula relation)

From the ancestor relation between nodes in a formula tree we define a *subformula relation* between formula occurrences in a natural way. If a node in the formula tree represents φ and an *ancestor* node represents ψ, then ψ is a *subformula occurrence* of φ. We also say the ψ is an *ancestor* of φ. If the node is an immediate ancestor node, then ψ is an *immediate subformula occurrence*, or an *immediate ancestor*, of φ. Any instance of ψ, i.e. any occurrence of ψ where the free variables have been replaced by terms, is also considered to be a subformula and an ancestor of φ. ⊣

Example 2.20 $(Gf^1_{16}u^1_{17})^1_{19}$ is a subformula occurrence of the formula occurrence in Example 2.18. It represents the formula tree for $(Gxy)^1_{19}$ together with the substitution $\{x/f^1_{16}, x/u^1_{17}\}$

2.5 THE CALCULUS $\mathsf{LK}^{\mathsf{ce}}$

In the skeletons we have seen so far, two types of formula copying have taken place. First, there is the *explicit* copying of a γ-formula in a γ-inference. The introduced copy is a different occurrence from the occurrence which is being copied. Second, there is the *implicit* copying which takes place in all β-inferences. All our rules have been context-sharing, so all extra formula occurrences of a conclusion of a β-inference have been copied into both of the premisses. In some ways, two implicit copies of a formula occurrence should be more closely related than two explicit copies. This is formulated in the notion of *contextual equivalence*.

2.21 Definition (Contextual equivalence)

Two different formula occurrences are *contextually equivalent* if they are representations of the same formula tree; equivalently, if they have the same index pair. ⊣

The calculus $\mathsf{LK}^{\mathsf{ce}}$ (ce stands for contextual equivalence) is essentially the same as introduced by Waaler [44]. The δ-rules of this system, call them δ^{ce}-rules, are like δ^+ [28], with the exception that contextually equivalent δ-formulas introduce identical Skolem functions. A certain reuse of Skolem functions is thus built into the calculus, but it not as strong as the δ^{+^+}-rule [10]; on any given branch, all introduced Skolem function symbols will be different. To obtain this, we utilize the index pairs provided by the underlying formula trees. If a δ-formula has the index pair ^n_m then a Skolem function symbol f^n_m is introduced[1], and if a γ-formula has the index pair

[1] A closer approximation to the δ^{+^+}-rule could be obtained by skipping the copy numbers of the Skolem function symbols altogether.

^n_m then an instantiation variable u^n_m is introduced.

2.22 Definition (Instantiation terms, revisited)

The sets $\mathcal{U}$ and $\mathcal{S}$ of instantiation variables and Skolem functions, respectively, will from now on be assumed to have the following form.
$\mathcal{U} = \{u^n_m \mid m, n \in \mathbb{N}\}$ and $\mathcal{S} = \{f^n_m \mid m, n \in \mathbb{N}\}$. A Skolem function f^n_m with no arguments will be written a^n_m. $\dashv$

2.23 Definition (The calculus $\mathsf{LK^{ce}}$)

The δ^{ce}-rules are given below. If the principal formula occurrence is $(\forall x \varphi^n_k)^n_m$ and $\vec{u}$ is a sequence of all instantiation variables occurring in $(\forall x \varphi^n_k)^n_m$, then the Skolem term $f^n_m(\vec{u})$ is introduced and substituted for the variable x in φ^n_k.

$$\frac{\Gamma \vdash \varphi[x/f^n_m(\vec{u})]^n_k, \Delta}{\Gamma \vdash (\forall x \varphi^n_k)^n_m, \Delta}\ \mathsf{R}\forall \qquad \frac{\Gamma, \varphi[x/f^n_m(\vec{u})]^n_k \vdash \Delta}{\Gamma, (\exists x \varphi^n_k)^n_m \vdash \Delta}\ \mathsf{L}\exists$$

The γ-rules for $\mathsf{LK^{ce}}$ are given below. If the principal formula occurrence is $(\forall x \varphi^n_k)^n_m$, then the instantiation variable u^n_m is introduced and substituted for the variable x in φ^n_k. Also, the principal formula occurrence is copied by incrementing *all* copy numbers in the underlying formula tree by one.

$$\frac{\Gamma, (\forall x \varphi^{n+1}_k)^{n+1}_m, \varphi[x/u^n_m]^n_k \vdash \Delta}{\Gamma, (\forall x \varphi^n_k)^n_m \vdash \Delta}\ \mathsf{L}\forall \qquad \frac{\Gamma \vdash (\exists x \varphi^{n+1}_k)^{n+1}_m, \varphi[x/u^n_m]^n_k, \Delta}{\Gamma \vdash (\exists x \varphi^n_k)^n_m, \Delta}\ \mathsf{R}\exists$$

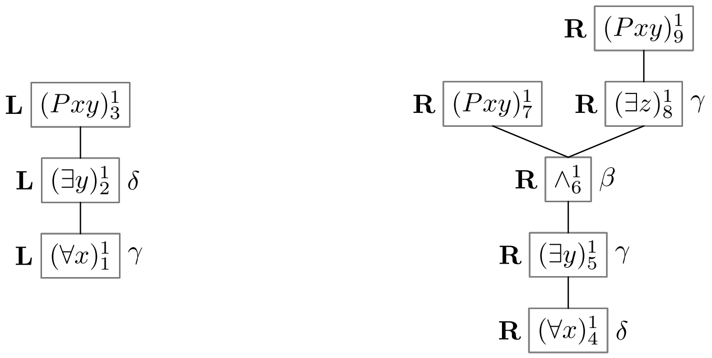
The closure condition is the same as for LK^δ: An $\mathsf{LK^{ce}}$-skeleton π is closable if there is a substitution σ such that all leaf sequents in π are σ-axioms. The proof will, like before, be denoted $\langle \pi, \sigma \rangle$. $\dashv$

Example 2.24 Again, here are proofs of $\forall x Px \vdash \forall x Px$, this time in $\mathsf{LK^{ce}}$. With all its index pairs displayed the sequent is of the form $(\forall x (Px)^1_3)^1_1 \vdash (\forall x (Px)^1_2)^1_4$. Below to the left is a proof with the same order of rule applications as in Example 1.36 and 2.9. To the right is a proof with the reversed order.

$$\begin{array}{c} u^1_1 \mapsto a^1_2 \\ \dfrac{\dfrac{\forall x Px, Pu^1_1 \vdash Pa^1_2}{\forall x Px \vdash Pa^1_2}\ u^1_1}{\forall x Px \vdash \forall x Px}\ a^1_2 \end{array} \qquad \begin{array}{c} u^1_1 \mapsto a^1_2 \\ \dfrac{\dfrac{\forall x Px, Pu^1_1 \vdash Pa^1_2}{\forall x Px, Pu^1_1 \vdash \forall x Px}\ a^1_2}{\forall x Px \vdash \forall x Px}\ u^1_1 \end{array}$$

We see that the rule dependencies between $\mathsf{L}\forall$ and $\mathsf{R}\forall$, which we had in LK and LK^{δ}, are gone with $\mathsf{LK}^{\mathsf{ce}}$. This is because of the liberalized δ-rules, which disregard other free variables than those occurring in the principal formula.

Example 2.25 The LK^{δ}-skeleton from Example 2.15 contains an inference (labeled $\delta_{g(u,v,w)}$) which introduces a Skolem term with three arguments. In $\mathsf{LK}^{\mathsf{ce}}$, the corresponding Skolem term will only have one argument. We will do this example in detail, since it is referred to later on. The formula trees of the formula occurrences in the root sequent is given below.



The $\mathsf{LK}^{\mathsf{ce}}$-proof is given below. The root sequent has been labeled with the occurrence numbers, and copy numbers are shown as superscript in the appropriate places. The closing unifier is shown above the branches.

$$u_1^2 \mapsto f_2^1(a_4^1),\ u_8^1 \mapsto f_2^2(a_4^1(f_2^1))$$

$$\dfrac{Pu_1^2 f_2^2(u_1^2), Pu_1^1 f_2^1(u_1^1) \vdash Pu_5^1 \boxed{u_8^1}}{Pu_1^2 \boxed{f_2^2(u_1^2)}, Pu_1^1 f_2^1(u_1^1) \vdash \exists z P u_5^1 z}\ u_8^1$$

$$\dfrac{Pu_1^2 \boxed{f_2^2(u_1^2)}, Pu_1^1 f_2^1(u_1^1) \vdash \exists z P u_5^1 z}{\exists y P \boxed{u_1^2} y, Pu_1^1 f_2^1 u_1^1 \vdash \exists z P u_5^1 z}\ f_2^2(u_1^2)$$

$$\dfrac{\exists y P \boxed{u_1^2} y, Pu_1^1 f_2^1 u_1^1 \vdash \exists z P u_5^1 z}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1 u_1^1 \vdash \exists z P u_5^1 z}\ u_1^2$$

$$u_1^1 \mapsto a_4^1,\ u_5^1 \mapsto f_2^1(a_4^1)$$

$$\dfrac{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash P a_4^1 u_5^1 \qquad (\forall x \exists y Pxy)^2, Pu_1^1 f_2^1 u_1^1 \vdash \exists z P u_5^1 z}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash P a_4^1 \boxed{u_5^1} \wedge \exists z P \boxed{u_5^1} z}$$

$$\dfrac{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash P a_4^1 \boxed{u_5^1} \wedge \exists z P \boxed{u_5^1} z}{(\forall x \exists y Pxy)^2, Pu_1^1 \boxed{f_2^1(u_1^1)} \vdash \exists y (P a_4^1 y \wedge \exists z Pyz)}\ u_5^1$$

$$\dfrac{(\forall x \exists y Pxy)^2, Pu_1^1 \boxed{f_2^1(u_1^1)} \vdash \exists y (P a_4^1 y \wedge \exists z Pyz)}{(\forall x \exists y Pxy)^2, \exists y P \boxed{u_1^1} y \vdash \exists y (P a_4^1 y \wedge \exists z Pyz)}\ f_2^1(u_1^1)$$

$$\dfrac{(\forall x \exists y Pxy)^2, \exists y P \boxed{u_1^1} y \vdash \exists y (P a_4^1 y \wedge \exists z Pyz)}{\forall x \exists y Pxy \vdash \exists y (P \boxed{a_4^1} y \wedge \exists z Pyz)}\ u_1^1$$

$$\dfrac{\forall x \exists y Pxy \vdash \exists y (P \boxed{a_4^1} y \wedge \exists z Pyz)}{(\underset{1}{\forall x} \underset{2}{\exists y} \underset{3}{Pxy})^1 \vdash \underset{4}{\forall x} \underset{5}{\exists y} (\underset{7}{Pxy} \underset{6}{\wedge} \underset{8}{\exists z} \underset{9}{Pyz})^1}\ a_4^1$$

2.6 Completeness of $\mathsf{LK}^{\mathsf{ce}}$

2.26 Theorem (Completeness of $\mathsf{LK}^{\mathsf{ce}}$)

All valid sequents are $\mathsf{LK}^{\mathsf{ce}}$-provable. ⊣

PROOF. Let $\Gamma \vdash \Delta$ be a valid sequent. By completeness of LK^{δ}, there is an LK^{δ}-proof π of $\Gamma \vdash \Delta$. By simulating this skeleton in $\mathsf{LK}^{\mathsf{ce}}$, a skeleton π' and a closing substitution σ are obtained. □

2.7 Soundness of $\mathsf{LK}^{\mathsf{ce}}$

We have only defined truth for *closed* formulas, and the strategy used for proving soundness of LK^{δ} does not work for $\mathsf{LK}^{\mathsf{ce}}$. Surely, we have a grounding lemma for $\mathsf{LK}^{\mathsf{ce}}$ (proven just like for LK^{δ}), but if we replace all occurrences of instantiation variables in an $\mathsf{LK}^{\mathsf{ce}}$-proof with *ground* terms, then a local soundness lemma, analogous to those for LK (Lemma 1.38) and LK^{δ} (Lemma 2.12), cannot be shown.

Example 2.27 To the left is an $\mathsf{LK}^{\mathsf{ce}}$-proof from Example 2.24, and to the right is the object obtained by replacing u_1^1 with a_2^1.

$$u_1^1 \mapsto a_2^1$$

$$\dfrac{\dfrac{\forall x Px, Pu_1^1 \vdash Pa_2^1}{\forall x Px, Pu_1^1 \vdash \forall x Px}\,a_2^1}{\forall x Px \vdash \forall x Px}\,u_1^1 \qquad\qquad \dfrac{\dfrac{\forall x Px, Pa_2^1 \vdash Pa_2^1}{\forall x Px, Pa_2^1 \vdash \forall x Px}\,\times}{\forall x Px \vdash \forall x Px}\,a_2^1$$

The inference marked with $\times$ is not sound in general. It is an instance of

$$\dfrac{\Gamma \vdash \varphi[x/a], \Delta}{\Gamma \vdash \forall x \varphi, \Delta}\,\times$$

where there is no restriction, like the eigenparameter condition, on the term a. Specifically, a can occur in the conclusion, which it does above.

There are two main ways of showing that $\mathsf{LK}^{\mathsf{ce}}$ is sound; (1) semantically, by introducing semantics for instantiation terms and showing that all rules preserve falsifiability upwards, and (2) syntactically, by showing that any $\mathsf{LK}^{\mathsf{ce}}$-proof can be transformed into an LK^{δ}-proof.

2.7.1 The semantical approach

To show soundness with the semantical approach, we first need to know what it means for a formula with instantiation terms to be true.

Recall that a structure M for a language $\mathcal{L}$ consist of a domain D and an interpretation function $(\cdot)^M$, which interprets all function and predicate symbols in $\mathcal{L}$. To interpret formulas with instantiation terms, the first step is to interpret the Skolem functions appropriately. We do so by extending $\mathcal{L}$ to a language $\mathcal{L}^{\mathsf{sko}}$ in which all the Skolem functions are added as function symbols. There is no harm in doing this, since the set of Skolem functions is assumed to be disjoint from the set of function symbols in $\mathcal{L}$. For the purpose of this section, all structures of consideration will be $\mathcal{L}^{\mathsf{sko}}$-structures.

The next step consists of interpreting the instantiation variables appropriately.

2.28 Definition (Variable assignment)

A *variable assignment* h for an $\mathcal{L}^{\mathsf{sko}}$-structure M is a function from instantiation variables to elements in $|M|$. ⊣

2.29 Definition (Interpretation)

An $\mathcal{L}^{\mathsf{sko}}$-structure M together with a variable assignment h for M, written $\langle M, h\rangle$, enables us to interpret all terms in the following way:

- $f(t_1, \ldots, t_n)^{\langle M,h\rangle} = f^M(t_1^{\langle M,h\rangle}, \ldots, t_n^{\langle M,h\rangle})$, where f is a function symbol in $\mathcal{L}^{\mathsf{sko}}$ (thus including all Skolem functions)

- $u^{\langle M,h\rangle} = h(u)$, for all instantiation variables u

By using the above interpretation of terms, it is now easy to define truth for formulas with instantiation variables. That a formula φ is true in a structure M under a variable assignment h, written $\langle M, h\rangle \models \varphi$, is defined analogous to Definition 1.12 (just replace M with $\langle M, h\rangle$). The only part of the truth definition that needs to be changed is the base case, which is now like this:

- For atomic formulas: $\langle M, h\rangle \models P(t_1, \ldots, t_n)$ if $(t_1^{\langle M,h\rangle}, \ldots, t_n^{\langle M,h\rangle}) \in P^M$

Some notation: $\langle M, h\rangle \models \Gamma$ means that φ is *true* in $\langle M, h\rangle$ for all formulas φ in Γ. $\langle M, h\rangle \models^{\perp} \Delta$ means that φ is *false* in $\langle M, h\rangle$ for all formulas φ in Δ. If $\langle M, h\rangle \models \varphi$, we say that φ is true in $\langle M, h\rangle$. ⊣

2.7.2 Why local soundness does not hold for $\mathsf{LK}^{\mathsf{ce}}$

For the previous calculi, we have shown local soundness lemmas; that all inferences preserve falsifiability upwards, meaning that if the conclusion is falsifiable, then at least one of the premisses is falsifiable. It is instructive to see precisely how such a local soundness lemma, as formulated for LK and LK^{δ}, fails to hold for $\mathsf{LK}^{\mathsf{ce}}$. The purpose of this discussion is to illustrate an essential feature of many free variable calculi; that different occurrences of the *same* instantiation variable in a skeleton, most typically in different branches, must be treated equally. For instance, all occurrences of an instantiation variable must be interpreted in the same way by a variable assignment, and all occurrences of an instantiation variable should be substituted with the same element. Calculi for which this is the case are called *rigid* free variable calculi.

The next natural step is to provide notions of validity and falsifiability for sequents in which instantiation variables occur. The simplest way of doing this is by means of a countermodel.

2.30 Definition (Countermodel)

If M is a $\mathcal{L}^{\mathsf{sko}}$-structure such that for all variable assignments h it is the case that $\langle M, h\rangle$ makes all formulas in Γ true and all formulas in Δ false, then M is a *countermodel* for the sequent $\Gamma \vdash \Delta$. A sequent is *valid* if it has no countermodel. $\dashv$

A more compact formulation which says exactly the same is: For all h, $\langle M, h\rangle \models \Gamma$ and $\langle M, h\rangle \models^{\perp} \Delta$.

Indirectly, instantiation variables occurring in antecedents should be interpreted universally and instantiation variables in succedents should be interpreted existentially. To see this, consider the following examples.

Example 2.31 $\Gamma, Pa \vdash Pu, \Delta$ is a valid sequent.

PROOF. It is sufficient to show that the sequent has no countermodel. Suppose that M is a countermodel. Then, for all variable assignments h, $\langle M, h\rangle$ must make Pa true and Pu false. But a variable assignment h which sends u to a^M makes both Pa true and Pu true in $\langle M, h\rangle$. This contradicts the assumption that M is a countermodel, and since M was arbitrary, no countermodel can exist for the sequent.

Here is a skeleton which generates such a sequent. Notice that u occurs in the succedent, and that it should have an existential interpretation.

$$\dfrac{\dfrac{\Gamma, Pa \vdash Pu, \exists x Px, \Delta}{\Gamma, Pa \vdash \exists x Px, \Delta}\,u}{\Gamma, \exists x Px \vdash \exists x Px, \Delta}\,a$$

□

Example 2.32 By the same reasoning as above, both $Pu \vdash Pa$ and $Pu \vdash Pu$ are valid sequents.

Remark. Validity for sequents could have been defined without going via countermodels, but the definition would not be so transparent. The definition goes like this: A sequent $\Gamma \vdash \Delta$ is valid if for all structures M there is a variable assignment h such that if all formulas in Γ are true in $\langle M, h\rangle$, then there is a formula in Δ which is true in $\langle M, h\rangle$.

Remark. As one might think, the sequent $\varphi_1, \ldots, \varphi_n \vdash \psi_1, \ldots, \psi_m$ does *not* correspond to the universal closure of $(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_m)$, which is $\forall\vec{x}\forall\vec{y}((\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_m))$, where $\vec{x}$ and $\vec{y}$ are the instantiation variables occurring in the φ_i's and the ψ_i's, respectively. Rather, it corresponds to $\forall\vec{x}(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \exists\vec{y}(\psi_1 \vee \ldots \vee \psi_m)$. To see this, consider the sequent with this formula in the succedent. After finitely many rule applications (in particular; $\mathsf{R}\!\rightarrow$, $\mathsf{L}\forall$, $\mathsf{L}\wedge$, $\mathsf{R}\exists$ and $\mathsf{R}\vee$) the sequent $\varphi_1, \ldots, \varphi_n \vdash \psi_1, \ldots, \psi_m$ is in the leaf node.

Example 2.33 The following sequents are falsifiable:

- $Pa, Qb \vdash Pu \wedge Qu$ - (Consider the structure M with domain $\{a, b\}$, in which Pa and Qb are true and Pb and Qa are false. For all assignments h, the formula $Pu \wedge Qu$ is false in $\langle M, h\rangle$.)
- $Pu \vee Qu \vdash Pa, Qb$ - (Consider a structure with the same domain, but where Pa and Qb are false and Pb and Qa are true. For all assignments h, the formula $Pu \vee Qu$ is true in $\langle M, h\rangle$.)

Let us apply $\mathsf{R}\wedge$ to the uppermost sequent in the above example to see why an analogous local soundness lemma *cannot* be shown for $\mathsf{LK^{ce}}$.

$$\dfrac{Pa, Qb \vdash Pu \quad Pa, Qb \vdash Qu}{Pa, Qb \vdash Pu \wedge Qu}\,\mathsf{R}\wedge$$

If considered separately, the two leaf sequents are valid, since there are no countermodels for either of them. This is exactly the place where a local soundness lemma fails to hold; the leaf nodes *cannot* be treated separately, because the instantiation variable u occurs in both of them. In this case, u is called a *rigid* variable. Logically, it must be a placeholder for the same element in both branches. Said differently, the quantification cannot be considered branchwise, but must span across the branches.

We now turn our attention to the right way to do it.

2.34 Definition (Open branch, semantically)

Let π be a skeleton and β a branch in a skeleton. Also, let β^+ consist of all formula occurrences in any antecedent of β and β^- consist of all formula occurrences in any succedent of β. Given a structure M and a variable assignment h, we say that β is an *open branch* of π if $\langle M, h\rangle$ makes all formulas in β^+ true and $\langle M, h\rangle$ makes all formulas in β^- false. $\dashv$

2.35 Definition (Countermodel for a skeleton)

A *countermodel for a skeleton* is a structure M such that for all variable assignments h there is an open branch in the skeleton. $\dashv$

Since instantiation variables function as placeholders, different variable assignments might give different open branches. If a skeleton has a countermodel, all we know is that for each variable assignment, there is an open branch.

Example 2.36 Reconsider the skeleton above.

$$\dfrac{Pa, Qb \vdash Pu \quad Pa, Qb \vdash Qu}{Pa, Qb \vdash Pu \land Qu}\ \mathsf{R}\land$$

The structure M given in Example 2.33 is a countermodel for the skeleton. For all variable assignments h there is an open branch in the skeleton. There are only two possibilities for the value of u. If $h(u) = a$, then the right branch is open; all antecedent formula occurrences are true and all succedent formula occurrences are false. Likewise, if $h(u) = b$, then the left branch is open.

This is an example of a general principle: all rules of $\mathsf{LK}^{\mathsf{ce}}$ preserve the existence of a countermodel for a skeleton.

2.37 Lemma (Countermodel preservation) Let π be a skeleton and let β be a branch in π with a leaf sequent $\Gamma \vdash \Delta$. Let r be an inference with $\Gamma \vdash \Delta$ as conclusion, and let π' be the skeleton obtained by adding the premiss(es) of r above $\Gamma \vdash \Delta$. Claim: If π has a countermodel, then π' has a countermodel. $\dashv$

Proof. If r is a one-premiss inference, call the new branch β_1; if r is a two-premiss inference, call the new branches β_1 and β_2. The new skeleton π' must be of one of these forms:

$$\begin{array}{c} \dfrac{\Gamma' \vdash \Delta'}{\Gamma \vdash \Delta}\, r \\ \vdots\, \beta \\ \pi \end{array} \qquad\qquad \begin{array}{c} \dfrac{\Gamma' \vdash \Delta' \quad \Gamma'' \vdash \Delta''}{\Gamma \vdash \Delta}\, r \\ \vdots\, \beta \\ \pi \end{array}$$

First, assume that M is a countermodel for π. The cases in the proof correspond to which type of inference r is. If r is not a δ-inference, then the countermodel for π' will be M.

Pick an arbitrary variable assignment h. Then, by assumption, there is an open branch in π. If this branch is different from β, then it must also be an open branch in π', and we are done. So, assume without loss of generality that the open branch in π is β. Then, β_1 (or β_2) will be an open branch of π'.

$\mathsf{L}\wedge$: Suppose β_1 is obtained by extending the leaf node of β in the following way:

$$\dfrac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}\, \mathsf{L}\wedge$$

Since β is open in π, we have that $\langle M, h\rangle \models \Gamma, \varphi\wedge\psi$ and $\langle M, h\rangle \models^{\perp} \Delta$. It follows that $\langle M, h\rangle \models \varphi$ and $\langle M, h\rangle \models \psi$, so β_1 is an open branch in π'.

$\mathsf{L}\vee$: Suppose β_1 and β_2 are obtained by extending the leaf node of β in the following way:

$$\dfrac{\Gamma, \varphi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta}\, \mathsf{L}\vee$$

Since β is open in π, we have that $\langle M, h\rangle \models \Gamma, \varphi\vee\psi$ and $\langle M, h\rangle \models^{\perp} \Delta$. Then, either $\langle M, h\rangle \models \varphi$ or $\langle M, h\rangle \models \psi$, which makes either β_1 or β_2 open.

$\mathsf{R}\wedge$, $\mathsf{R}\vee$, $\mathsf{R}\rightarrow$, $\mathsf{L}\rightarrow$, $\mathsf{R}\neg$, $\mathsf{L}\neg$ are similar.

$\mathsf{L}\forall$: Suppose β_1 is obtained by extending the leaf node of β in the following way:

$$\dfrac{\Gamma, \forall x\varphi, \varphi[x/u] \vdash \Delta}{\Gamma, \forall x\varphi \vdash \Delta}\, \mathsf{L}\forall$$

Since β is an open branch, we have that $\langle M, h\rangle \models \forall x\varphi$. Claim: $\langle M, h\rangle \models \varphi[x/u]$. Assume $h(u) = d$, where d is in the domain of

M. Since $\langle M, h\rangle$ makes $\forall x\varphi$ true, by the truth definition it is the case that $\langle M, h\rangle \models [x/\overline{d}]$. Since $h(u) = d$, we have that $\langle M, h\rangle \models \varphi[x/u]$. (This is easily established by induction on the length of formulas.) Then β_1 is a an open branch of π'.

R$\exists$: Similar.

If the rule used to obtain β_1 was L$\exists$ or R$\forall$, we must construct a new countermodel M' for π'.

L$\exists$: Suppose β_1 is obtained by extending the leaf node of β in the following way:

$$\frac{\Gamma, \varphi[x/f(u_1,\ldots,u_n)] \vdash \Delta}{\Gamma, \exists x\varphi \vdash \Delta}\ \mathsf{L}\exists$$

Let M' be a structure which is *identical to* M except for the interpretation of the Skolem function f, which should be interpreted in the following way. Let $a_1,\ldots,a_n$ be n elements in the domain of M', and let h be a variable assignment which maps u_i to a_i for $1 \leq i \leq n$. If $\langle M, h\rangle \models \exists x\varphi$, then there is an element d in the domain of M' such that $\langle M, h\rangle \models \varphi[x/\overline{d}]$. In that case, let $f^M(a_1,\ldots,a_n) = d$. If not, let $f^M(a_1,\ldots,a_n) = d'$ for an arbitrary d' in the domain of M'.

Claim: M' is a countermodel for π'. Again, pick an arbitrary variable assignment h. Notice, that h is also a variable assignment for M, since M and M' have the same domain. Then, there is a branch β' in π which is open with respect to $\langle M, h\rangle$. If β' does not contain the Skolem function f, then it must also be open in π' with respect to $\langle M', h\rangle$, since the two structures interpret all other symbols identically. But, if β' *does* contain the Skolem function f (this subtlety is caused by the fact that the Skolem function f might already occur in π), then there are two cases. (1) If β' is identical to β, then β_1 will be an open branch of π'. This is guaranteed by the construction of M'. In that case, $\langle M', h\rangle \models \exists x\varphi$ and the interpretation of f gives that $\langle M', h\rangle \models \varphi[x/f(u_1,\ldots,u_n)]$. (2) If β' is *a different* branch than β in which f occurs, then the situation is a bit more complex. The branch β' is open with respect to $\langle M, h\rangle$, but since the M' interprets f differently, we cannot be sure that β' is open with respect to $\langle M', h\rangle$. Nevertheless, the construction of M' ensures that f is interpreted *in the right way*, and we use this fact to show that there indeed is a branch in π' which is open with respect to $\langle M', h\rangle$. First, there must be contextually equivalent formula occurrences in β' of both $\exists x\varphi$ and $\varphi[x/f(u_1,\ldots,u_n)]$. (Otherwise, f would not be in β'.) These must also be true in $\langle M', h\rangle$, since M and M' only differ with respect to the interpretation of f. Let r' be the inference in β' in which $\exists x\varphi$ is principal. By induction on the structure of the skeleton *above* r', it it easily shown that there must be an open branch with respect to $\langle M', h\rangle$.

Remark. The open branch found in this last step is not necessarily identical to β'. This is because M and M' might interpret f differently. For example, if $\forall x\varphi$ is of the form $\exists x(Pxu \lor Qxu)$:

$$\frac{\Gamma, Pf(u)u \lor Qf(u)u \vdash \Delta}{\Gamma, \exists x(Pxu \lor Qxu) \vdash \Delta}\ f(u)$$

It is possible that M interprets f such that $Pf(u)u$ is false for a given variable assignment h, and that the construction of M' results in an interpretation of f such that $Pf(u)u$ is *true* under the the same variable assignment.

□

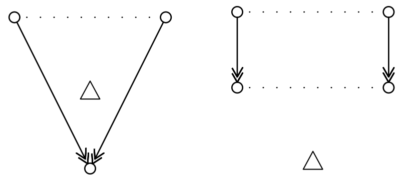
2.38 Theorem (Soundness of $\mathsf{LK}^{\mathsf{ce}}$)

Let $\langle \pi, \sigma\rangle$ be a proof in $\mathsf{LK}^{\mathsf{ce}}$ of a sequent $\Gamma \vdash \Delta$. Then $\Gamma \vdash \Delta$ is valid. ⊣

Proof. Suppose that $\Gamma \vdash \Delta$ is not valid. Then, the skeleton consisting of only $\Gamma \vdash \Delta$ has a countermodel M. By repeated applications of the countermodel preservation lemma (Lemma 2.37), M must be a countermodel for π. By definition of a countermodel, there must be an open branch β in π for any variable assignment h. By Lemma 2.11 there is a *ground* substitution σ' which closes π. Let h be the variable assignment which sends an instantiation variable u to the interpretation of $u\sigma'$ in M, i.e. $h(u) = (u\sigma')^M$. (Since $u\sigma'$ is a ground term and M is a $\mathcal{L}^{\mathsf{sko}}$-structure, $(u\sigma')^M$ is well-defined.) Since every leaf sequent is a σ'-axiom, there cannot be an open branch in π with respect to $\langle M, h\rangle$. Then, $\Gamma \vdash \Delta$ must be valid. □

Chapter 3

A change of perspective



Permutation variants

For each inference in a skeleton, there is exactly one principal formula occurrence. We can thus relate inferences in a skeleton by means of how their principal formula occurrences are related.

This is a change of perspective. The relations introduced below are *not* between the nodes in a skeleton, which are sequents, but between the *inferences* in a skeleton. Inferences contain more information than sequents; each inference has a type, a principal formula occurrence, one or two active formula occurrences and a context.

3.1 Relations between inferences

3.1 Definition (Contextual equivalence)

Two different inferences r and s are *contextually equivalent*, written $r \sim s$, if their principal formula occurrences are contextually equivalent. The set consisting of r together with contextually equivalent inferences of r is denoted $[r]$. $\dashv$

Remark. In the definition, the inferences are assumed to be *different*. For reasons that will become apparent later, it is not convenient that an inference is contextually equivalent to itself.

3.2 Definition (Immediate ancestor)

The inference r is an *immediate ancestor* of the inference s, written $r \gg s$, if (1) the principal formula occurrence of r is an immediate ancestor of the principal formula occurrence of s, *and* (2) r and s are inferences in the same branch of the skeleton. Equivalently, the principal formula occurrence of r has the same index pair as an active formula occurrence of s. (The ancestor relation is defined similarly.) $\dashv$

3.3 Definition (Substitution ordering)

Let π be a skeleton and σ a closing unifier. Assume without loss of generality that σ is *minimal* in the sense that every binding in the support of σ is necessary in order to close all branches of π. (A binding is necessary for the closure if the result of taking the binding out of the support is that some branch is not closed. If this is not the case, we can remove the bindings which are not necessary.) Then, $r_u \sqsupset r_f$ holds if is the case that:

(1) r_u and r_f are two inferences in the *same branch* β of π

(2) r_u introduces the instantiation variable u

(3) r_f introduces the Skolem function f

(4) $u\sigma = f(\vec{t})$, for some sequence of terms $\vec{t}$

(5) u and f both occur in the atomic formula which closes β

We say that the $\sqsupset$ is the substitution ordering induced from σ. $\dashv$

3.4 Definition (Conforming)

Let $\langle \pi, \sigma \rangle$ be a proof. π *conforms to* $\sqsupset$, if for all inferences r and s in π, such that $r \sqsupset s$, it is the case that r is above s. $\dashv$

Remark. A conforming skeleton corresponds to the eigenparameter condition of LK.

Example 3.5

$$
\dfrac{\dfrac{\dfrac{\dfrac{Pua \vdash Pbv}{Pua \vdash \exists y Pby}\,v}{\forall x Pxa \vdash \exists y Pby}\,u}{\forall x Pxa \vdash \forall x \exists y Pxy}\,b}{\exists y \forall x Pxy \vdash \forall x \exists y Pxy}\,a
$$

Let a, b, u, v denote the inferences in the skeleton. Then, $u \gg a$ holds because the principal formula of u ($\forall x Pxa$) is ancestor to the principal formula of a ($\exists y \forall x Pxy$). By the same reasoning $v \gg b$ holds. Let $\sigma = \{u/b, v/a\}$ be a closing unifier. Then $u \sqsupset b$ and $v \sqsupset a$ hold. The skeleton conforms to σ, since u is above b and v is above a.

3.6 Lemma (Soundness of $\mathsf{LK}^{\mathsf{ce}}$- Part I) Let $\langle \pi, \sigma \rangle$ be a proof of $\Gamma \vdash \Delta$ in $\mathsf{LK}^{\mathsf{ce}}$ such that π conforms to $\sqsupset$. Then, $\Gamma \vdash \Delta$ is valid. $\dashv$

Proof. Since π conforms to $\sqsupset$ it is possible to show that $\Gamma \vdash \Delta$ is valid by induction on the length of $\pi\sigma'$ for a ground substitution σ'. This is precisely analogous to the local soundness proof for both LK (Lemma 1.38) and LK^{δ} (Lemma 2.12). The conformity of σ ensures that the Skolem terms do not occur in the conclusion when they are introduced. They function properly as witnesses for satisfiability (in $\mathsf{R}\exists$-inferences) and unsatisfiability (for $\mathsf{L}\forall$-inferences). $\Box$

3.7 Definition ($\gg$-tree)

If r is an inference in a skeleton, then the *$\gg$-tree* rooted in r is the least tree T such that (1) $r \in T$ and (2) for all $s \in T$, if $s' \gg s$, then $s' \in T$. $\dashv$

Observation. Let T be the $\gg$-tree rooted in r such that φ is principal in r. Let T' be the tree where the nodes are exactly the sets $[s]$, where s is an inference in T, such that $[s]$ is an immediate ancestor of $[s']$ if and only if $s \gg s'$. Each node represents an equivalence class of contextually equivalent inferences. Then T' is isomorphic to the formula tree of φ in the following way: Each node $[s]$ in T', such that ψ^n_m is principal in s, is associated with the node in the formula tree representing ψ^n_m. This node will have the same principal type as the inference type of s, and the polarity will correspond to whether ψ^n_m occurs in the antecedent or the succedent. This enables us to speak of $\gg$-trees and formula trees in a similar way. More importantly, for each $\gg$-tree, there is a unique formula tree. Conversely, for each formula tree there can be many $\gg$-trees in a skeleton, but only one up to contextual equivalence.

3.2 Diagram representations

A rewarding part of this work has consisted in developing an intuitive way of representing derivations, inferences and relations between inferences in order to reason about them effectively. By observing that many important properties of derivations (such as the fulfillment of an eigenparameter condition) could be expressed by means of relations between inferences, as opposed to relations between formulas or sequents, it was natural to develop a diagram language which could display the essential aspects and dependencies in a derivation. The diagram language enables us to give an abstract and compact representation of a derivation; furthermore, operations on these representations correspond directly to operations on the represented derivations. It enables us to visualize complex structures and abstract away from the not so essential details of a derivation. It turns out that reasoning about these representations, and not about the represented structures in all detail, is very fruitful. It is not uncommon for mathematicians to employ such visual structures in their reasoning; by reasoning about simpler structures, avoiding an unnecessary detail overload, it becomes possible to gain insight and prove important facts about the real, more complex, structures. Hoping that it will increase clarity and display our logical intuitions, the diagram representations of skeletons will be provided as often as possible after they are introduced.

A skeleton is composed out of inferences, instances of rules. We will now represent skeletons at a higher level of abstraction by diagrams in which the diagram labels denote inferences and arrows denote relations between inferences.

The following are diagram labels:

α-inference:	$\circ$	γ-inference:	u or u^n_m
β-inference:	$\triangle$	δ-inference:	$f(\vec{u})$, $f^n_m(\vec{u})$, a, a^n_m

If a γ-inference introduces the instantiation variable u, then we use u as a label representing this inference. If a δ-inference introduces the instantiation term $f(\vec{u})$, where f is the Skolem function, then we use $f(\vec{u})$ as a label representing this inference.

The following are the diagram arrows:

immediate ancestor relation	$\twoheadrightarrow$
contextual equivalence relation	$\cdots\cdots$
substitution ordering	$\dashrightarrow$

A diagram representation of the inference r being an immediate ancestor of an inference s.

$$\begin{tikzcd} r \arrow[d, two heads] \\ s \end{tikzcd} \qquad (r \gg s)$$

A diagram representation of the inference r being contextually equivalent to an inference s due to a splitting caused by a β-rule.

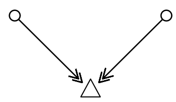
$$\begin{array}{ccc} r & \cdots\cdots\cdots & s \\ & \triangle & \end{array} \qquad (r \sim s)$$

A diagram representation of two inferences, r_u and s_f, being $\sqsupset$-related.

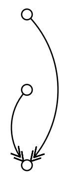
$$\begin{tikzcd} f \\ u \arrow[u, dashed] \end{tikzcd} \qquad\qquad (r_u \sqsupset s_f)$$

The diagram labels enable us to distinguish between to types of splitting of branches in $\gg$-trees; one type of splitting due to α-rules and one due to β-rules, which additionally split the branches of the skeleton.

$$\frac{\dfrac{\Gamma \vdash \varphi', \Delta}{\Gamma \vdash \varphi, \Delta} \qquad \dfrac{\Gamma \vdash \psi', \Delta}{\Gamma \vdash \psi, \Delta}}{\Gamma \vdash \varphi \wedge \psi, \Delta}$$



$$\frac{\dfrac{\dfrac{\Gamma \vdash \varphi', \psi', \Delta}{\Gamma \vdash \varphi', \psi, \Delta}}{\Gamma \vdash \varphi, \psi, \Delta}}{\Gamma \vdash \varphi \vee \psi, \Delta}$$



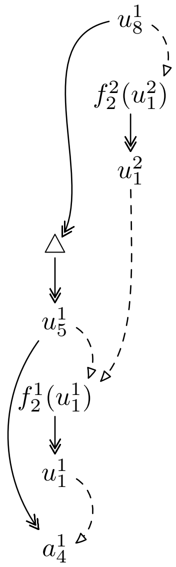
3.8 Definition (Balanced skeleton)

A *balanced skeleton* is a skeleton where the following condition holds for all inferences r and s: If $r \sim s$ and $r' \gg r$, then there is an s' such that $s' \gg s$ and $s \sim s'$. A diagram representation:

$$\begin{tikzcd} r' \arrow[d, two heads] & \\ r \arrow[r, dotted, no head] & s \end{tikzcd} \quad \text{only if there is a } s' \text{ such that} \quad \begin{tikzcd} r' \arrow[d, two heads] \arrow[r, dotted, no head] & s' \arrow[d, two heads] \\ r \arrow[r, dotted, no head] & s \end{tikzcd} \qquad \dashv$$

Remark. The notion of balanced skeletons is the same as the notion of *bisimilarity*, as found in [14] and often used in modal logic. What is described above is exactly what it means for two structures to be bisimilar, namely that there exists a bisimulation between the two structures.

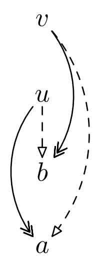
To get a feeling for the diagrams, here is a diagram representation for the $\mathsf{LK}^{\mathsf{ce}}$-proof in Example 2.25. Notice that all arrows point downwards. Thus, the skeleton is conforming. This property of the diagrams corresponds to the eigenparameter condition. We say that the proof in question is *LK-like*. Also, observe that each label representing a γ-inference only has one arrow going downwards out from it. Later, we will see a calculus where this is not always the case.



The skeleton from Example 3.5, together with its corresponding diagram, illustrating that the skeleton is conforming.

$$u/b, v/a$$

$$\dfrac{\dfrac{\dfrac{\dfrac{Pua \vdash Pbv}{Pua \vdash \exists y Pby}\,v}{\forall x Pxa \vdash \exists y Pby}\,u}{\forall x Pxa \vdash \forall x \exists y Pxy}\,b}{\exists y \forall x Pxy \vdash \forall x \exists y Pxy}\,a$$



3.3 PERMUTATIONS

The study of permutations for sequent calculi goes back to Kleene [33]. Some background knowledge will be presupposed here. For precise definitions and more detailed expositions, see Waaler [44] or Troelstra and Schwichtenberg [42]. Despite this, many examples will be given, and the reader should be able to get a fairly good understanding of permutations from these.

Intuitively, a *permutation* of a skeleton, called a *permutation variant*, is a skeleton which differs only in the *order* of rule applications. Two permutation variants have exactly the same leaf sequents.

What is important to point out here is that only *symmetrical* permutation schemes are considered. The class of balanced and variable-sharing (see definition below) skeletons was identified in [44], where it was shown that leaf sequents of such skeletons correspond precisely to paths through matrices. (See Bibel [12] or Wallen [46] for further information about matrices.)

3.9 Definition (Variable-sharing skeleton)

A skeleton is *variable-sharing* if all contextually equivalent γ-inferences introduce the *same* instantiation variables. A skeleton is *variable-pure* if all γ-inferences introduce *different* instantiation variables. ⊣

Example 3.10 LK^δ generates variable-pure skeletons and $\mathsf{LK}^{\mathsf{ce}}$ generates variable-sharing skeletons. This depends solely on how the γ-rules are defined. It is possible to define $\mathsf{LK}^{\mathsf{ce}}$ such that it generates variable-pure skeletons, but then symmetrical permutation schemes would not be applicable.

With LK and LK^δ there are rule dependencies (cnf. Example 1.36 for LK and Example 2.9 for LK^δ) which make it impossible to permute freely between inferences. In particular, a γ-inference ($\mathsf{L}\forall$ or $\mathsf{R}\exists$) in a skeleton might depend on a δ-inference ($\mathsf{R}\forall$ or $\mathsf{L}\exists$), and then it is necessary for the δ-inference to occur *below* the γ-inference in any given branch. With $\mathsf{LK}^{\mathsf{ce}}$, this is not the case anymore (cnf. Example 2.24).

Even if we are not going into the details of permutations, it is instructive to see how symmetrical permutation schemes can be applied. The next three examples are all illustrations of symmetrical permutation schemes. Notice that the permutation variants always have identical leaf sequents.

Example 3.11 The two proofs in Example 2.24 are permutation variants; here, two one-premiss inferences change order.

Example 3.12 A one-premiss inference changes order with a two-premiss inference:

$$\dfrac{\dfrac{Pu \vdash \varphi}{\forall x Px \vdash \varphi}\,\mathsf{L}\forall \qquad \dfrac{Pu \vdash \psi}{\forall x Px \vdash \psi}\,\mathsf{L}\forall}{\forall x Px \vdash \varphi \land \psi}\,\mathsf{R}\land \qquad\qquad \dfrac{\dfrac{Pu \vdash \varphi \qquad Pu \vdash \psi}{Pu \vdash \varphi \land \psi}\,\mathsf{R}\land}{\forall x Px \vdash \varphi \land \psi}\,\mathsf{L}\forall$$

Example 3.13 Two two-premiss inferences change order:

$$\dfrac{\dfrac{\varphi_1 \vdash \psi_1 \qquad \varphi_2 \vdash \psi_1}{\varphi_1 \lor \varphi_2 \vdash \psi_1}\,\mathsf{L}\lor \qquad \dfrac{\varphi_1 \vdash \psi_2 \qquad \varphi_2 \vdash \psi_2}{\varphi_1 \lor \varphi_2 \vdash \psi_2}\,\mathsf{L}\lor}{\varphi_1 \lor \varphi_2 \vdash \psi_1 \land \psi_2}\,\mathsf{R}\land$$

$$\dfrac{\dfrac{\varphi_1 \vdash \psi_1 \quad \varphi_1 \vdash \psi_2}{\varphi_1 \vdash \psi_1 \wedge \psi_2}\,\mathsf{R}\wedge \qquad \dfrac{\varphi_2 \vdash \psi_1 \quad \varphi_2 \vdash \psi_2}{\varphi_2 \vdash \psi_1 \wedge \psi_2}\,\mathsf{R}\wedge}{\varphi_1 \vee \varphi_2 \vdash \psi_1 \wedge \psi_2}\,\mathsf{L}\vee$$

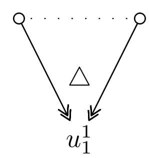
A useful lemma to have when reasoning about permutations is the following.

3.14 Lemma (Permutation) Let π be a balanced skeleton with a non-atomic formula occurrence φ in the root sequent such that φ has an ancestor somewhere in the skeleton. Then, there is a permutation variant of π where φ is principal in the lowermost inference. $\dashv$

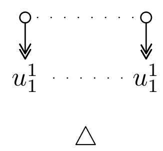
PROOF. (See Lemma 2.14 in Waaler [44].) Briefly, the proof there goes like this: Since φ has an ancestor in π, it must be principal somewhere in π. There must be a set of contextually equivalent inferences which all have φ as the principal formula occurrence. All of these inferences must occur in different branches. By repeatedly choosing the inferences which are furthest away from the root and permuting downwards according to permutation schemes, a skeleton such that φ is principal in the lowermost inference is obtained. $\square$

Example 3.15 The following two skeletons are permutation variants. In the first skeleton, the γ-inference has been applied before the β-inference; in the second skeleton, it is the other way around. To the right are the corresponding diagrams.

$$\dfrac{\dfrac{\dfrac{\dfrac{Pu_1^1, Qu_1^1 \vdash \varphi}{Pu_1^1 \wedge Qu_1^1 \vdash \varphi} \qquad \dfrac{Pu_1^1, Qu_1^1 \vdash \psi}{Pu_1^1 \wedge Qu_1^1 \vdash \psi}}{Pu_1^1 \wedge Qu_1^1 \vdash \varphi \wedge \psi}\,\mathsf{R}\wedge}{\forall x(Px \wedge Qx) \vdash \varphi \wedge \psi}\,u_1^1}{}$$



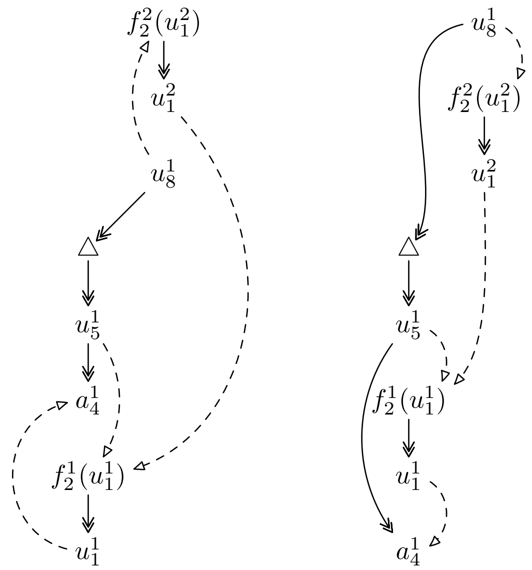
$$\dfrac{\dfrac{\dfrac{Pu_1^1, Qu_1^1 \vdash \varphi}{Pu_1^1 \wedge Qu_1^1 \vdash \varphi}}{\forall x(Px \wedge Qx) \vdash \varphi}\,u_1^1 \qquad \dfrac{\dfrac{Pu_1^1, Qu_1^1 \vdash \psi}{Pu_1^1 \wedge Qu_1^1 \vdash \psi}}{\forall x(Px \wedge Qx) \vdash \psi}\,u_1^1}{\forall x(Px \wedge Qx) \vdash \varphi \wedge \psi}\,\mathsf{R}\wedge$$



Example 3.16 (More complex.) The skeleton in Example 2.25 has an important property; it is *conforming* with respect to the closing substitution. Below is a permutation variant of this skeleton which is *not* conforming. The permutation variant has exactly the same leaf sequents as the original skeleton.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash Pa_4^1 u_5^1}}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash Pa_4^1 u_5^1} \quad \dfrac{\dfrac{\dfrac{Pu_1^2 f_2^2(u_1^2), Pu_1^1 f_2^1(u_1^1) \vdash Pu_5^1 u_8^1}{\exists y Pu_1^2 y, Pu_1^1 f_2^1(u_1^1) \vdash Pu_5^1 u_8^1}\, f_2^2(u_1^2)}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash Pu_5^1 u_8^1}\, u_1^2}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash \exists z Pu_5^1 z}\, u_8^1}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash Pa_4^1 u_5^1 \wedge \exists z Pu_5^1 z}}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash \exists y (Pa_4^1 y \wedge \exists z Pyz)}\, u_5^1}{(\forall x \exists y Pxy)^2, Pu_1^1 f_2^1(u_1^1) \vdash \forall x \exists y (Pxy \wedge \exists z Pyz)}\, a_4^1}{(\forall x \exists y Pxy)^2, \exists y Pu_1^1 y \vdash \forall x \exists y (Pxy \wedge \exists z Pyz)}\, f_2^1(u_1^1)}{(\underset{1}{\forall} x \underset{2}{\exists} y \underset{3}{P} xy)^1 \vdash \underset{4}{\forall} x \underset{5}{\exists} y (\underset{7}{P} xy \underset{6}{\wedge} \underset{8}{\exists} z \underset{9}{P} yz)^1}\, u_1^1
$$

The leftmost diagram represents the permutation variant, and the rightmost diagram represents the original skeleton. Notice that the arrows from u_1^1 to a_4^1 and from u_8^1 to $f_2^2(u_1^2)$ go *upwards*. This is what makes the skeleton *non-conforming*. Also, observe that the permutation variant is obtained by *moving* a_4^1 upwards two inferences and by moving u_8^1 downwards two inferences.



3.4 Cycle elimination

In this section we will approximate a syntactical and proof-theoretical soundness proof for $\mathsf{LK}^{\mathsf{ce}}$.

3.17 Definition (Reduction ordering)

Let $\langle \pi, \sigma \rangle$ be a proof in $\mathsf{LK}^{\mathsf{ce}}$. We introduce a new relation $\underset{\sim}{\gg}$ between inferences, which is the transitive closure of $\gg$, *with a twist* added to it. Let $\gg^{+}$ be the transitive (but not reflexive) closure of $\gg$. Then, $r \underset{\sim}{\gg} r'$ holds if either (1) $r \gg^{+} r'$ or (2) there is an inference r'' such that $r \gg^{+} r''$ and $r'' \sim r'$ (r'' is contextually equivalent to r'). Diagrammatically, $r \underset{\sim}{\gg} r'$ holds if:

Either (1)
$$\begin{tikzcd} r \arrow[d, twoheadrightarrow, "+"] \\ r' \end{tikzcd}$$
or (2)
$$\begin{tikzcd} r \arrow[d, twoheadrightarrow, "+"] & \\ r'' \arrow[r, dotted, no head] & r' \end{tikzcd}$$

The transitive (but not reflexive) closure of ($\underset{\sim}{\gg} \cup \sqsupset$) gives the *reduction ordering* $\rhd$. ($\sqsupset$ is the substitution ordering induced from σ). We say that $\rhd$ is the *reduction ordering* induced from σ. $\dashv$

3.18 Definition (Cycle)

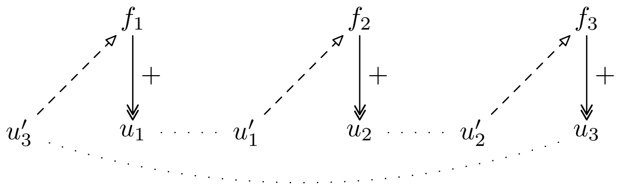
Let $\langle \pi, \sigma \rangle$ be a proof in $\mathsf{LK}^{\mathsf{ce}}$. A $\rhd$-*cycle* is a finite sequence of inferences $r_1, \ldots, r_n$, for $n \geq 2$, such that $r_1 \rhd r_2, \ldots, r_n \rhd r_1$. We say that $\rhd$ *contains* the cycle $r_1, \ldots, r_n$. It is *cycle-free* if it does not contain a cycle. $\dashv$

Convention. Since $\underset{\sim}{\gg}$ is transitive we will from now on assume that each $\rhd$-cycle is a finite sequence of inferences $r_1, \ldots, r_n$ such that $r_1 \underset{\sim}{\gg} r_2$, $r_2 \sqsupset r_3$, $\ldots$, $r_{n-1} \underset{\sim}{\gg} r_n$, $r_n \sqsupset r_1$. Additionally, since $r \sqsupset r'$ holds only if r is a γ-inference and r' is a δ-inference, it is also safe to assume that if $s \underset{\sim}{\gg} s'$ holds, for s and s' in a cycle, then s is a δ-inference and s' is a γ-inference. Furthermore, each $\underset{\sim}{\gg}$-step can be *decomposed* into exactly one $\gg^{+}$-step and one $\sim$-step.

3.19 Definition (Cycle length)

The *length of a cycle* is the number of γ-inferences in it; equivalently the number of δ-inferences in it. (By the above assumption, there are equally many γ- and δ-inferences in a cycle.) $\dashv$

Since we have diagram representations for both $\gg$ and $\sqsupset$, and $\rhd$ is defined from these, it is possible to give a diagram representation of a $\rhd$-cycle. We use the fact that one $\underset{\sim}{\gg}$-step in a cycle can be decomposed into one $\gg^{+}$-step and one $\sim$-step. Below is a diagram representation of a cycle of length 3, where $f_1 \gtrsim u'_1$, $u'_1 \sqsupset f_2$, $f_2 \gtrsim u'_2$, $u'_2 \sqsupset f_3$, $f_3 \gtrsim u'_3$ and $u'_3 \sqsupset f_1$.



3.20 Definition (Well-founded)

A relation $>$ is *well-founded* on a set A if for all non-empty subsets B of A, there is a $>$-minimal element in B, i.e. an element r such that there is no element s in B such that $r > s$. $\dashv$

3.21 Lemma $\rhd$ is well-founded if and only if it is cycle-free. $\dashv$

Proof ($\Rightarrow$). If $\rhd$ contains a cycle, then it is obviously not well-founded. The set of all inferences in the given skeleton does not have a $\rhd$-minimal element. $\square$

Proof ($\Leftarrow$). If $\rhd$ is not well-founded, then there is a subset of inferences in π for which there is no $\rhd$-minimal inference. For every inference r in the subset it is possible to find an inference s in the subset such that $r \rhd s$. But, then $\rhd$ contains a cycle. $\square$

3.22 Lemma (Soundness of $\mathsf{LK}^{\mathsf{ce}}$- Part II) Let $\langle \pi, \sigma \rangle$ be a proof in $\mathsf{LK}^{\mathsf{ce}}$ such that the reduction ordering $\rhd$ induced from σ is cycle-free. Then, there is a permutation variant π which conforms to $\sqsupset$. $\dashv$

Remark. Together with part I (Lemma 3.6), this lemma gives that every sequent with a proof $\langle \pi, \sigma \rangle$ such that $\rhd$ is cycle-free, is valid.

Proof. By induction on the sub-skeletons of π. Initially, label all inferences in π *black*.

Basic step: By Lemma 3.21, for the set of inferences in π, there must be a $\rhd$-minimal element r. Assume that φ is the principal formula occurrence of r. By the permutation lemma (Lemma 3.14), there is a permutation variant of π which has φ as the principal formula occurrence in the lowermost inference, i.e. r is the lowermost inference of the permutation variant. Label the inference r *white*.

Induction hypothesis: The $\sqsupset$-conformity property holds for all white inferences, i.e. if r and s are white inferences such that $r \sqsupset s$, then r is above s in the skeleton.

Induction step: Let π' be a sub-skeleton such that its lowermost sequent is the premiss of a white inference. Observe that all inferences in π' are black and that all inferences below its lowermost sequent are white. By Lemma 3.21, for the set of inferences in π', there must be a $\rhd$-minimal element r. By the permutation lemma there is a permutation variant π'' of π' in which r is the lowermost inference. This must also be a permutation variant of the main skeleton (of which π' is a sub-skeleton). Since all inferences in π' were black, no white inference was touched in order to obtain π'', so we can apply the induction hypothesis. Label the inference r *white*. Claim: The $\sqsupset$-conformity property still holds for all white inferences. The only possibility for this not to be the case is if $s \sqsupset r$, for some white inference s. But then, since $s \rhd r$, the inference s could not have been $\rhd$-minimal when it was labeled white. $\square$

So far, so good. We have now established that $\mathsf{LK}^{\mathsf{ce}}$ is a sound calculus, *provided that the induced* $\rhd$ *is cycle-free.* The only problem is that $\rhd$ can contain cycles, as the reader might have guessed by now. When $\rhd$ contains a cycle, there is no conforming permutation variant, and the arguments above can not be applied to show that the sequent in question is valid.

The soundness proof in Waaler [44] fails to recognize the existence of such cycles and does not work as it stands. The error lies in Lemma 2.16 in that article, which says: "*The reduction ordering* $\rhd$ *induced by a ground, homogeneous substitution is well-founded.*" The reduction ordering defined there is essentially[1] the same as the one defined here.

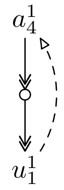
Before a sketch for a cycle elimination proof is given, we will go through some examples of cycles and how to eliminate them in full detail.

Example 3.23 (Cycle - main example I)

Let φ_1^n abbreviate: $(\underset{1}{\exists} x(\underset{3}{P}x \underset{2}{\rightarrow} \underset{4}{\forall} y \underset{5}{P}y))_1^n$. A proof in $\mathsf{LK}^{\mathsf{ce}}$ is given below.

[1] There are two main differences between $\rhd$ in Waaler [44] and our $\rhd$: (1) The $\rhd$ in [44] also relates inferences which do *not* occur in the same branch. (2) The $\rhd$ in [44] also relates γ-inferences with each other. The choice of reduction ordering in this thesis avoids (1) because this allows for a more fine-grained discussion of cycles. (2) is avoided since only ground substitutions are considered and the essence of a conforming skeleton is not affected by whether two γ-inferences are $\sqsupset$-related.

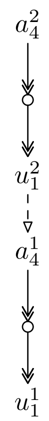
$$\cfrac{\cfrac{\cfrac{\begin{array}{c}u_1^1/a_4^1\\ Pu_1^1 \vdash Pa_4^1, \varphi_1^2\end{array}}{Pu_1^1 \vdash \forall yPy, \varphi_1^2}\;a_4^1}{\vdash Pu_1^1 \to \forall yPy, \varphi_1^2}}{\vdash \exists x(Px \to \forall yPy)}\;u_1^1$$



There is a $\triangleright$-cycle of length one, since $u_1^1 \sqsupset a_4^1$ and $a_4^1 \gg u_1^1$.

If we want to eliminate the cycle from this example, there is only one thing to do; expand the formula occurrence φ_1^2. After one application of $\mathsf{R}\exists$ and one application of $\mathsf{R}{\to}$, the leaf node is $Pu_1^1, Pu_1^2 \vdash Pa_4^1, \forall yPy, \varphi_1^3$. Now it is possible to close the skeleton with the binding $\{u_1^2/a_4^1\}$, which results in a cycle-free proof. The skeleton resulting from additionally applying $\mathsf{L}\forall$is given below, together with a diagram representation.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\begin{array}{c}u_1^2/a_4^1\\ Pu_1^1, Pu_1^2 \vdash Pa_4^1, Pa_4^2, \varphi_1^3\end{array}}{Pu_1^1, Pu_1^2 \vdash Pa_4^1, \forall yPy, \varphi_1^3}\;a_4^2}{Pu_1^1 \vdash Pa_4^1, Pu_1^2 \to \forall yPy, \varphi_1^3}}{Pu_1^1 \vdash Pa_4^1, \varphi_1^2}\;u_1^2}{Pu_1^1 \vdash \forall yPy, \varphi_1^2}\;a_4^1}{\vdash Pu_1^1 \to \forall yPy, \varphi_1^2}}{\vdash \exists x(Px \to \forall yPy)}\;u_1^1$$



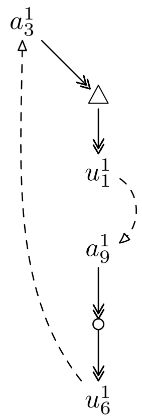
Example 3.24 (Cycle - main example II)

Let φ_1^n abbreviate $(\underset{1}{\forall} x(\underset{3}{\forall} x \underset{4}{Q}x \underset{2}{\to} \underset{5}{P}x))_1^n$.

Let ψ_6^n abbreviate $\underset{6}{\exists} x(\underset{8}{Q}x \underset{7}{\to} \underset{9}{\forall} x \underset{10}{P}x)_6^n$.

A proof of the sequent $\varphi_1^1 \vdash \psi_6^1$, in which the principal formula occurrences are boxed, is given below. To the right is the corresponding diagram representation, which demonstrates the cycle.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}u_6^1/a_3^1\\ \varphi_1^2, Qu_6^1 \vdash \psi_6^2, Pa_9^1, Qa_3^1\end{array}}{\varphi_1^2, Qu_6^1 \vdash \psi_6^2, Pa_9^1, \boxed{\forall x Qx}}\,a_3^1 \qquad \begin{array}{c}u_1^1/a_9^1\\ Pu_1^1, \varphi_1^2, Qu_6^1 \vdash \psi_6^2, Pa_9^1\end{array}}{\varphi_1^2, \boxed{\forall xQx \to Pu_1^1}, Qu_6^1 \vdash \psi_6^2, Pa_9^1}}{\boxed{\forall x(\forall xQx \to Px)}, Qu_6^1 \vdash \psi_6^2, Pa_9^1}\,u_1^1}{\forall x(\forall xQx \to Px), Qu_6^1 \vdash \psi_6^2, \boxed{\forall xPx}}\,a_9^1}{\forall x(\forall xQx \to Px) \vdash \psi_6^2, \boxed{Qu_6^1 \to \forall xPx}}}{\forall x(\forall xQx \to Px) \vdash \boxed{\exists x(Qx \to \forall xPx)}}\,u_6^1
$$



There is a $\triangleright$-cycle of length two: $u_6^1 \sqsupset a_3^1$, $a_3^1 \gtrsim u_1^1$, $u_1^1 \sqsupset a_9^1$ and $a_9^1 \gtrsim u_6^1$.

The rightmost leaf sequent is closed by the binding $\{u_1^1/a_9^1\}$. If this binding is removed (taken out of the support of the substitution), then the cycle would be eliminated. In order to close the skeleton without this binding, but *with* the other bindings in the support as they are, we can introduce a new instantiation variable u_1^2 and a new binding $\{u_1^2/a_9^1\}$ for all the leaf sequents in which u_1^1 occur. To get the variable u_1^2, we must expand φ_1^2. The result of doing this for the rightmost leaf sequent is given below.

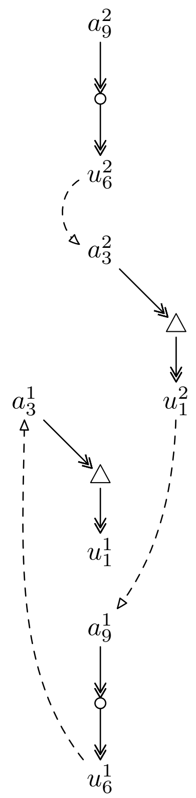
$$
\dfrac{\dfrac{\dfrac{Pu_1^1, \varphi_1^3, Qu_6^1 \vdash \psi_6^2, Pa_9^1, Qa_3^2}{Pu_1^1, \varphi_1^3, Qu_6^1 \vdash \psi_6^2, Pa_9^1, \boxed{\forall xQx}}\,a_3^1 \qquad \begin{array}{c}u_1^2/a_9^1\\ Pu_1^2, Pu_1^1, \varphi_1^3, Qu_6^1 \vdash \psi_6^2, Pa_9^1\end{array}}{Pu_1^1, \varphi_1^3, \boxed{\forall xQx \to Pu_1^2}, Qu_6^1 \vdash \psi_6^2, Pa_9^1}}{Pu_1^1, \boxed{\varphi_1^2}, Qu_6^1 \vdash \psi_6^2, Pa_9^1}\,u_1^2
$$

The rightmost branch in the new skeleton is closed by $\{u_1^2/a_9^1\}$, but the other new branch is *not* closed; the variable u_6^1 is already bound to a_3^1, so it cannot be sent to a_3^2, and the variable u_1^1 is not supposed to be in the support of the new substitution. In the same spirit, we can expand ψ_6^2 in order to get a new variable u_6^2, which can be sent to a_3^2. Then a cycle-free skeleton is obtained. (See the diagram representation – variant 1.)

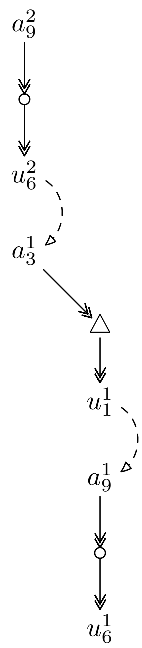
$$\dfrac{\dfrac{\overset{u_6^2/a_3^2}{Pu_1^1, \varphi_1^3, Qu_6^1, Qu_6^2 \vdash \forall x Px, \psi_6^3, Pa_9^1, Qa_3^2}}{Pu_1^1, \varphi_1^3, Qu_6^1 \vdash \boxed{Qu_6^2 \to \forall x Px}, \psi_6^3, Pa_9^1, Qa_3^2}}{Pu_1^1, \varphi_1^3, Qu_6^1 \vdash \boxed{\psi_6^2}, Pa_9^1, Qa_3^2}\ u_6^2$$

Another way of eliminating the cycle consists of removing the binding $\{u_6^1/a_3^1\}$ from the cycle, introducing the binding $\{u_6^2/a_3^1\}$ and expanding ψ_6^2 in the leftmost sequent of the original skeleton, since that is the leaf sequent in which u_6^1 occurs. (See the diagram representation – variant 2.)

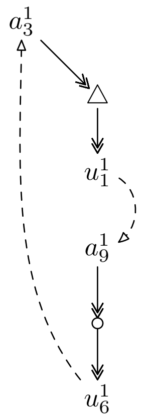
Variant 1.



Variant 2.

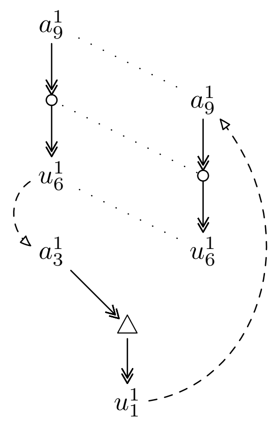


The original skeleton.



Below is a permutation variant of the first skeleton. The rules are applied in a different order, but the leaf nodes are identical. Again, the principal formula occurrences are boxed. Below is the corresponding diagram representation, which demonstrates the cycle.

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c} u_6^1/a_3^1 \\ \varphi_1^2, Qu_6^1 \vdash \psi_6^2, Pa_9^1, Qa_3^1 \end{array}}{\varphi_1^2, Qu_6^1 \vdash \psi_6^2, \boxed{\forall x P x}, Qa_3^1}\, a_9^1}{\varphi_1^2 \vdash \psi_6^2, \boxed{Qu_6^1 \to \forall x P x}, Qa_3^1}}{\varphi_1^2 \vdash \boxed{\exists x(Qx \to \forall x P x)}, Qa_3^1}\, u_6^1}{\varphi_1^2 \vdash \exists x(Qx \to \forall x P x), \boxed{\forall x Q x}}\, a_3^1
\qquad
\dfrac{\dfrac{\dfrac{\begin{array}{c} u_1^1/a_9^1 \\ Pu_1^1, \varphi_1^2, Qu_6^1 \vdash \psi_6^2, Pa_9^1 \end{array}}{Pu_1^1, \varphi_1^2, Qu_6^1 \vdash \psi_6^2, \boxed{\forall x P x}}\, a_9^1}{Pu_1^1, \varphi_1^2 \vdash \psi_6^2, \boxed{Qu_6^1 \to \forall x P x}}}{Pu_1^1, \varphi_1^2 \vdash \boxed{\exists x(Qx \to \forall x P x)}}\, u_6^1
}{
\dfrac{\varphi_1^2, \boxed{\forall x Q x \to P u_1^1} \vdash \exists x(Qx \to \forall x P x)}{\boxed{\forall x(\forall x Q x \to P x)} \vdash \exists x(Qx \to \forall x P x)}\, u_1^1
}
$$

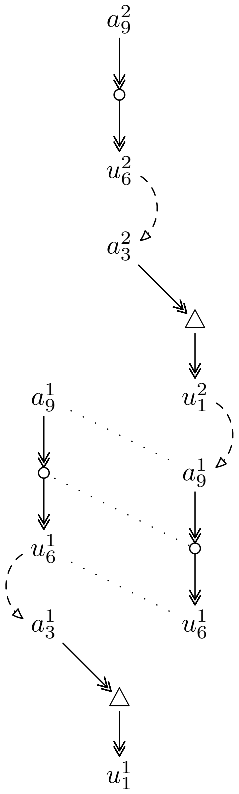


The same $\triangleright$-cycle of length two is found here: $u_1^1 \sqsupset a_9^1$, $a_9^1 \gtrsim u_6^1$, $u_6^1 \sqsupset a_3^1$ and $a_3^1 \gtrsim u_1^1$.

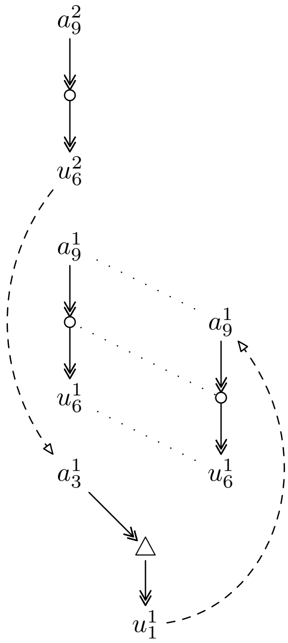
Remark. u_6^1 in the cycle represents the leftmost inference which introduces u_6^1, and a_9^1 represents the rightmost inference which introduces a_9^1. There is no harm in not specifying which, since $\sqsupset$ in this case unambiguously relates two and two inferences on a branch. If we inspect the decomposition of the cycle, this becomes clear: $u_1^1 \sqsupset a_9^1\mathbf{R}$, $a_9^1\mathbf{R} \gg^+ u_6^1\mathbf{R}$, $u_6^1\mathbf{R} \sim u_6^1\mathbf{L}$, $u_6^1\mathbf{L} \sqsupset a_3^1$ and $a_3^1 \gtrsim u_1^1$, where $\mathbf{L}$ and $\mathbf{R}$ denote whether the inference is in the left or the right branch of the skeleton.

By the exact same arguments as for the first skeleton it is possible to eliminate the cycle in the permutation variant in two ways, as indicated by the cycle-free diagrams below.

Variant 1.



Variant 2.



It is essential that cycles can be eliminated. The motivation for doing so should be clear from Lemma 3.22; every cycle-free proof has a conforming permutation variant.

The idea for a more abstract cycle elimination theorem is to "*break up*" cycles in the following way:

Let $\langle \pi, \sigma \rangle$ be an $\mathsf{LK}^{\mathsf{ce}}$-proof for which $\triangleright$ contains a cycle. Let r_u and r_f be two inferences in the cycle such that r_u introduces the instantiation variable u, r_f introduces the Skolem function f and $r_u \sqsupset r_f$. The substitution σ must send u to a term of the form $f(\vec{t})$. Let τ be the substitution $\sigma \setminus \{u/f(\vec{t})\}$; i.e. σ where u is *not* in the support. It should be possible to *extend* π to a skeleton π' and *extend* τ to a substitution τ' such that τ' closes π'. Assume that the principal formula occurrence of r is φ_m^n. Then, u is identical to u_m^n. The inference r also introduces a copy of φ_m^n, which means that in any leaf sequent of π where u_m^n occurs, there is a formula occurrence of φ_m^{n+k}, for some $k \geq 1$. By expanding this formula occurrence, a new instantiation variable u_m^{n+k} is introduced, which *does not occur in the support of* τ. Therefore, we are free to assign any value to u_m^{n+k}. In particular, we can extend τ such that u_m^{n+k} is sent to f.

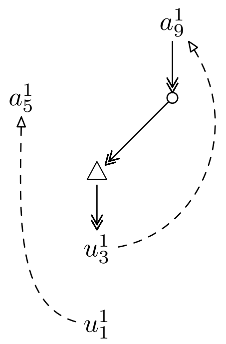
3.25 Conjecture (Cycle elimination)

For every proof $\langle \pi, \sigma \rangle$ there is an extension of π with a closing substitution σ' such that the reduction ordering $\rhd$ induced from σ' is cycle-free. $\dashv$

One essential obstacle is the following: When expanding a γ-formula occurrence φ in order to obtain a fresh instantiation variable, there might be β-type subformula occurrences of φ which split the skeleton into branches. In order to close all new branches under the condition that some instantiation variables should not be in the support of the closing substitution, it is sometimes necessary to expand other γ-formula occurrences as well.

Example 3.26 Let φ_3^n abbreviate: $(\underset{3}{\exists} x(\underset{5}{\forall} x \underset{6}{Qx} \underset{4}{\wedge} (\underset{8}{Px} \underset{7}{\rightarrow} \underset{9}{\forall} y \underset{10}{Py})))_1^n$.

$$
\begin{prooftree}
\AxiomC{u_1^1/a_5^1}
\noLine
\UnaryInfC{$\forall xQx, Qu_1^1 \vdash Qa_5^1, \varphi_3^2$}
\RightLabel{a_5^1}
\UnaryInfC{$\forall xQx, Qu_1^1 \vdash \forall xQx, \varphi_3^2$}
\AxiomC{u_3^1/a_9^1}
\noLine
\UnaryInfC{$\forall xQx, Qu_1^1, Pu_3^1 \vdash Pa_9^1, \varphi_3^2$}
\RightLabel{a_9^1}
\UnaryInfC{$\forall xQx, Qu_1^1, Pu_3^1 \vdash \forall yPy, \varphi_3^2$}
\UnaryInfC{$\forall xQx, Qu_1^1 \vdash Pu_3^1 \rightarrow \forall yPy, \varphi_3^2$}
\BinaryInfC{$\forall xQx, Qu_1^1 \vdash \forall xQx \wedge (Pu_3^1 \rightarrow \forall yPy), \varphi_3^2$}
\RightLabel{u_3^1}
\UnaryInfC{$\forall xQx, Qu_1^1 \vdash \exists x(\forall xQx \wedge (Px \rightarrow \forall yPy))$}
\RightLabel{u_1^1}
\UnaryInfC{$\forall xQx \vdash \exists x(\forall xQx \wedge (Px \rightarrow \forall yPy))$}
\end{prooftree}
$$



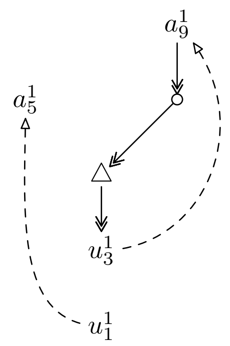
To eliminate the cycle by the strategy indicated above it is necessary to expand *both* the occurrence of φ_3^2 and the occurrence of $\forall xQx$.

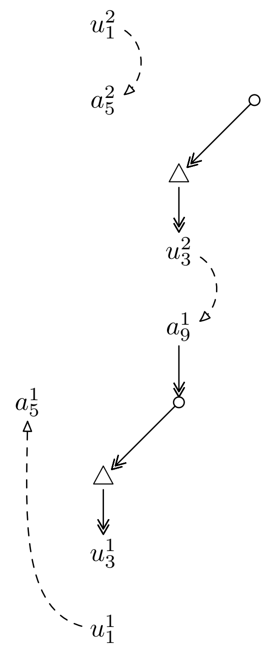
$$\dfrac{\dfrac{\dfrac{\forall xQx, Qu_1^1, Pu_3^1 \vdash Pa_9^1, Qa_5^2, \varphi_3^3}{\forall xQx, Qu_1^1, Pu_3^1 \vdash Pa_9^1, \forall xQx, \varphi_3^3}\, a_5^2 \qquad \dfrac{\overset{u_3^2/a_9^1}{\forall xQx, Qu_1^1, Pu_3^1 Pu_3^2 \vdash Pa_9^1, \forall yPy, \varphi_3^3}}{\forall xQx, Qu_1^1, Pu_3^1 \vdash Pa_9^1, Pu_3^2 \to \forall yPy, \varphi_3^3}}{\forall xQx, Qu_1^1, Pu_3^1 \vdash Pa_9^1, \forall xQx \wedge (Pu_3^2 \to \forall yPy), \varphi_3^3}}{\forall xQx, Qu_1^1, Pu_3^1 \vdash Pa_9^1, \varphi_3^2}\, u_3^2$$

In the left leaf sequent above, the binding u_1^1/a_5^2 would close the branch, but we already have the binding u_1^1/a_5^1, and the binding u_3^1/a_9^1 would close, but u_3^1 should not be in the support of the closing substitution. It is necessary to expand $\forall xQx$ once more.

$$\dfrac{\overset{u_1^2/a_5^2}{\forall xQx, Qu_1^2, Qu_1^1, Pu_3^1 \vdash Pa_9^1, Qa_5^2, \varphi_3^3}}{\forall xQx, Qu_1^1, Pu_3^1 \vdash Pa_9^1, Qa_5^2, \varphi_3^3}\, u_1^2$$

Below are the diagrams for the original skeleton and the cycle-free skeleton.





This shows that it is not sufficient to take only the γ-formula occurrences in a $\triangleright$-cycle into consideration when cycles are eliminated.

Chapter 4

Uniform variable splitting

> *"Given a proof, how can we reduce it to a smaller proof by exploiting the symmetricity of its subparts?"*
>
> A. Carbone [17]

4.1 Rigid and universal variables

All calculi so far have had *rigid* instantiation variables; whenever we have applied a substitution to an instantiation variable, we have applied the substitution to all occurrences of this variable. There are nevertheless situations where this is not necessary, where two occurrences of the same variable can play two logically different roles. In particular, the the occurrences can be place-holders for two different elements.

Let us start with a simple example:

$$\dfrac{\dfrac{\dfrac{u/a}{\forall x Px, Pu \vdash Pa}}{\forall x Px, Pu \vdash \forall y Py}\,a \qquad \dfrac{\dfrac{u/b}{\forall x Px, Pu \vdash Pb}}{\forall x Px, Pu \vdash \forall z Pz}\,b}{\dfrac{\forall x Px, Pu \vdash \forall y Py \wedge \forall z Pz}{\forall x Px \vdash \forall y Py \wedge \forall z Pz}\,u}$$

For the calculi we have seen so far, it is not possible to close this skeleton at this stage. The instantiation variable u must be sent to *both* a and b in order to close it. In this situation, however, there is no harm in viewing Pu as a *universal formula* (see [7] or [9] for details), and allowing u to be assigned *both* the value a *and* the value b. In cases where this is permissible u can be viewed as a *universal variable*.

The situation does not improve by permuting the skeleton in order to get the δ-inferences *below* the γ-inferences:

$$\dfrac{\dfrac{\dfrac{\overset{u/a}{\forall x Px, Pu \vdash Pa}}{\forall x Px \vdash Pa}\,u}{\forall x Px \vdash \forall y Py}\,a \qquad \dfrac{\dfrac{\overset{u/b}{\forall x Px, Pu \vdash Pb}}{\forall x Px \vdash Pb}\,u}{\forall x Px \vdash \forall z Pz}\,b}{\forall x Px \vdash \forall y Py \wedge \forall z Pz}$$

Since our free variable calculi are all variable-sharing, the two contextually equivalent γ-inferences must introduce *the same* instantiation variable. In LK, however, we have the freedom of instantiating γ-inferences with any terms we wish:

$$\dfrac{\dfrac{\dfrac{\forall x Px, Pa \vdash Pa}{\forall x Px \vdash Pa}\,\gamma_a}{\forall x Px \vdash \forall y Py}\,\delta_a \qquad \dfrac{\dfrac{\forall x Px, Pb \vdash Pb}{\forall x Px \vdash Pb}\,\gamma_b}{\forall x Px \vdash \forall z Pz}\,\delta_b}{\forall x Px \vdash \forall y Py \wedge \forall z Pz}$$

We see that in LK, the term universe relevant for instantiation of γ-formulas can be bound branchwise; it is sufficient to instantiate γ-formulas with terms occurring on the same branch as the γ-formula. There is a clear advantage of LK over free variable calculi when it comes to restricting the search space effectively. A proof procedure which has to consider all terms in all branches of a skeleton naturally has a greater search space than calculi where the term universe can be bound branchwise.

On the other hand, free variable calculi with variable-sharing skeletons have very nice permutation properties, which LK does not have.

In his book Automated Theorem Proving [12], Bibel sketched a method for systematic splitting of variables, called *splitting by need* The system introduced in this chapter can be viewed as a refinement of this idea.[1]

[1] Bibel's [12] splitting by need is defined for matrix systems and clausal formulas only. The system introduced here is designed to work also for non-clausal formulas.

4.2 A calculus with uniform variable splitting

4.1 Definition (Splitting set)

A *splitting set* is a set of index pairs. ⊣

Each formula occurrence in a skeleton will now be assumed to have a splitting set attached to it. The splitting sets are considered to be part of first-order languages, not just meta-language devices used to talk about formula occurrences. The splitting set of a formula occurrence is initially (in the root sequent) empty, and index pairs are added to splitting sets when β-rules are applied. The purpose of these sets is to keep track of which β-formula occurrences that have split the skeleton into branches, in order to split the instantiation variables accordingly, as explained below.

Example 4.2 Here is an example of a formula occurrence with a splitting set attached to it and all index pairs shown:

$$(\exists y((Pu_1^2 y)_4^2 \wedge (Qu_1^2)_5^2)_3^2)_2^2\{{}^1_7, {}^1_8\}$$

We will assume that all skeletons of this calculus are balanced. It is not known at the time of writing what happens if this restriction is lifted.[2]

Empty splitting sets will usually not be shown, and instead of writing commas, like $\{{}^1_3, {}^1_5, {}^1_7\}$, we will write $\{{}^1_3\,{}^1_5\,{}^1_7\}$.

4.3 Definition (The rules of $\mathsf{LK^s}$)

The rules of $\mathsf{LK^s}$ (the s stands for *splitting*) is given in Figure 4.1. $\varphi^n_m S$ denotes the formula occurrence φ with index pair ${}^n_m$ and splitting set S. The quantifier rules are like those for $\mathsf{LK^{ce}}$, but with splitting sets added to all formula occurrences.

β-rules: $\Gamma \uplus^l_k$ denotes $\{\varphi^n_m S \cup \{{}^l_k\} \mid \varphi^n_m S \in \Gamma\}$, the set of formula occurrences in Γ where the index pair ${}^l_k$ has been added to the splitting sets. The β-rules split the skeleton into branches and add the corresponding index pairs to the splitting sets of *all other formula occurrences than the β-formula itself. If the β-formula in question is $(\varphi^n_k \wedge \psi^n_l)^n_m S$, then the index pairs ${}^n_k$ and ${}^n_l$ are called* dual. *The dual index pair of ${}^n_m$ will be denoted ${}^n_{\overline{m}}$.*

[2] The author believes that this is possible, and that a calculus with many interesting properties will emerge from doing so. However, additional constraints for *closing unifiers* are probably needed, and identifying exactly which criteria these should satisfy is not entirely trivial.

α-rules

$$\frac{\Gamma, \varphi^n_k S, \psi^n_l S \vdash \Delta}{\Gamma, (\varphi^n_k \wedge \psi^n_l)^n_m S \vdash \Delta}\ \mathsf{L}\wedge$$

$$\frac{\Gamma \vdash \varphi^n_k S, \psi^n_l S, \Delta}{\Gamma \vdash (\varphi^n_k \vee \psi^n_l)^n_m S, \Delta}\ \mathsf{R}\vee$$

$$\frac{\Gamma, \varphi^n_k S \vdash \psi^n_l S, \Delta}{\Gamma \vdash (\varphi^n_k \to \psi^n_l)^n_m S, \Delta}\ \mathsf{R}\!\to$$

$$\frac{\Gamma, \varphi^n_k S \vdash \Delta}{\Gamma \vdash (\neg\varphi^n_k)^n_m S, \Delta}\ \mathsf{R}\neg$$

$$\frac{\Gamma \vdash \varphi^n_k S, \Delta}{\Gamma, (\neg\varphi^n_k)^n_m S \vdash \Delta}\ \mathsf{L}\neg$$

β-rules

$$\frac{\Gamma \uplus {}^n_k \vdash \varphi^n_k S, \Delta \uplus {}^n_k \quad \Gamma \uplus {}^n_l \vdash \psi^n_l S, \Delta \uplus {}^n_l}{\Gamma \vdash (\varphi^n_k \wedge \psi^n_l)^n_m S, \Delta}\ \mathsf{R}\wedge$$

$$\frac{\Gamma \uplus {}^n_k, \varphi^n_k S \vdash \Delta \uplus {}^n_k \quad \Gamma \uplus {}^n_l, \psi^n_l S \vdash \Delta \uplus {}^n_l}{\Gamma, (\varphi^n_k \vee \psi^n_l)^n_m S \vdash \Delta}\ \mathsf{L}\vee$$

$$\frac{\Gamma \uplus {}^n_k \vdash \varphi^n_k S, \Delta \uplus {}^n_k \quad \Gamma \uplus {}^n_l, \psi^n_l S, \vdash \Delta \uplus {}^n_l}{\Gamma, (\varphi^n_k \to \psi^n_l)^n_m S \vdash \Delta}\ \mathsf{L}\!\to$$

δ-rules

$$\frac{\Gamma \vdash \varphi[x/f^n_m(\vec{u})]^n_k S, \Delta}{\Gamma \vdash (\forall x \varphi^n_k)^n_m S, \Delta}\ \mathsf{R}\forall$$

$$\frac{\Gamma, \varphi[x/f^n_m(\vec{u})]^n_k S \vdash \Delta}{\Gamma, (\exists x \varphi^n_k)^n_m S \vdash \Delta}\ \mathsf{L}\exists$$

γ-rules

$$\frac{\Gamma, (\forall x \varphi^{n+1}_k)^{n+1}_m S, \varphi[x/u^n_m]^n_k S \vdash \Delta}{\Gamma, (\forall x \varphi^n_k)^n_m S \vdash \Delta}\ \mathsf{L}\forall$$

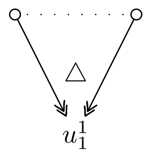
$$\frac{\Gamma \vdash (\exists x \varphi^{n+1}_k)^{n+1}_m S, \varphi[x/u^n_m]^n_k S, \Delta}{\Gamma \vdash (\exists x \varphi^n_k)^n_m S, \Delta}\ \mathsf{R}\exists$$

Figure 4.1: The rules of LK^{s}

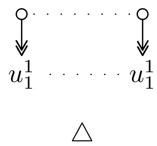
$\dashv$

Example 4.4 To get a feeling for the skeletons generated by this calculus, reconsider the permutation variants from Example 3.15, now with index pairs and splitting sets:

$$\dfrac{\dfrac{\dfrac{(Pu_1^1)\{^1_2\}, (Qu_1^1)\{^1_2\} \vdash \varphi_2^1}{(Pu_1^1 \land Qu_1^1)\{^1_2\} \vdash \varphi_2^1} \qquad \dfrac{(Pu_1^1)\{^1_3\}, (Qu_1^1)\{^1_3\} \vdash \psi_3^1}{(Pu_1^1 \land Qu_1^1)\{^1_3\} \vdash \psi_3^1}}{Pu_1^1 \land Qu_1^1 \vdash \varphi_2^1 \land \psi_3^1}\,\beta}{\forall x(Px \land Qx)_1^1 \vdash \varphi_2^1 \land \psi_3^1}\,\gamma_1^1$$



$$\dfrac{\dfrac{\dfrac{(Pu_1^1)\{^1_2\}, (Qu_1^1)\{^1_2\} \vdash \varphi_2^1}{(Pu_1^1 \land Qu_1^1)\{^1_2\} \vdash \varphi_2^1}}{\forall x(Px \land Qx)_1^1\{^1_2\} \vdash \varphi_2^1}\,\gamma_1^1 \qquad \dfrac{\dfrac{(Pu_1^1)\{^1_3\}, (Qu_1^1)\{^1_3\} \vdash \psi_3^1}{(Pu_1^1 \land Qu_1^1)\{^1_3\} \vdash \psi_3^1}}{\forall x(Px \land Qx)_1^1\{^1_3\} \vdash \psi_3^1}\,\gamma_1^1}{\forall x(Px \land Qx)_1^1 \vdash \varphi_2^1 \land \psi_3^1}\,\beta$$



The instantiation variable u_1^1 occurs in both leaf sequents, but *with different splitting sets* attached to the atomic formula occurrences in which it occurs. In the left branch, we find $(Qu_1^1)\{^1_2\}$ and in the right branch we find $(Qu_1^1)\{^1_3\}$. Speaking quite informally, since we have not yet defined the mechanism for splitting variables, *the calculus will allow us to view $u_1^1\{^1_2\}$ and $u_1^1\{^1_3\}$ as two different variables which can be bound to different terms.*

4.5 Lemma (Invariance lemma) The sets of leaf sequents are invariant under permutation. Any two permutation variants have identical leaf sequents. ⊣

PROOF. By induction on the number of primitive permutation steps performed to obtain the permutation variant. Each step is an application of a permutation scheme and results in a skeleton which has identical leaf sequents to the original. So, the final permutation variant must also have identical leaf sequents to the first skeleton. □

Example 4.6 Below is an instance of a permutation scheme for permuting a two-premiss inference over another two-premiss inference. It is analogous to Example 3.13.

$$\dfrac{\dfrac{\overset{\mathbf{1}}{\varphi_1^1\{\substack{1\\3}\} \vdash \psi_3^1\{\substack{1\\1}\}} \quad \overset{\mathbf{2}}{\varphi_2^1\{\substack{1\\3}\} \vdash \psi_3^1\{\substack{1\\2}\}}}{(\varphi_1^1 \lor \varphi_2^1)\{\substack{1\\3}\} \vdash \psi_3^1}\,\mathsf{L}\lor \qquad \dfrac{\overset{\mathbf{3}}{\varphi_1^1\{\substack{1\\4}\} \vdash \psi_4^1\{\substack{1\\1}\}} \quad \overset{\mathbf{4}}{\varphi_2^1\{\substack{1\\4}\} \vdash \psi_4^1\{\substack{1\\2}\}}}{(\varphi_1^1 \lor \varphi_2^1)\{\substack{1\\4}\} \vdash \psi_4^1}\,\mathsf{L}\lor}{\varphi_1^1 \lor \varphi_2^1 \vdash \psi_3^1 \land \psi_4^1}\,\mathsf{R}\land$$

$$\dfrac{\dfrac{\overset{\mathbf{1}}{\varphi_1^1\{\substack{1\\3}\} \vdash \psi_3^1\{\substack{1\\1}\}} \quad \overset{\mathbf{3}}{\varphi_1^1\{\substack{1\\4}\} \vdash \psi_4^1\{\substack{1\\1}\}}}{\varphi_1^1 \vdash (\psi_3^1 \land \psi_4^1)\{\substack{1\\1}\}}\,\mathsf{R}\land \qquad \dfrac{\overset{\mathbf{2}}{\varphi_2^1\{\substack{1\\3}\} \vdash \psi_3^1\{\substack{1\\2}\}} \quad \overset{\mathbf{4}}{\varphi_2^1\{\substack{1\\4}\} \vdash \psi_4^1\{\substack{1\\2}\}}}{\varphi_2^1 \vdash (\psi_3^1 \land \psi_4^1)\{\substack{1\\2}\}}\,\mathsf{R}\land}{\varphi_1^1 \lor \varphi_2^1 \vdash \psi_3^1 \land \psi_4^1}\,\mathsf{L}\lor$$

Both skeletons have identical leaf sequents, as indicated by the numbers above each.

4.7 Lemma (Permutation) Let π be a balanced sub-skeleton with a non-atomic formula occurrence $\varphi_m^n S$ in the root node such that $\varphi_m^n S$ has an ancestor somewhere in the skeleton. Then, there is a permutation variant of π which has $\varphi_m^n S$ as the principal formula occurrence in the lowermost inference. ⊣

PROOF. (Analogous to the proof of Lemma 3.14) □

The mechanism for splitting variables consists of identifying each instantiation variable by the splitting set of the surrounding formula occurrence. We will use a color metaphor and speak of *colored* variables, where splitting sets are the colors. Skeletons in which all leaf nodes in all branches are closed by a unifying substitution will still play an important part. To do this, substitutions must be defined for colored variables, and suitable conditions for these substitutions must be defined. Since all permutation variants of an underlying skeleton generate the same set of leaves, this can be done altogether at the level of terms.

A guiding intuition: The uniformity of the splitting sets should provide a machinery to split variables maximally *without losing sight of logical dependencies* between these.

4.8 Definition (Color)

A *color* is a splitting set occurring in a leaf sequent of a skeleton. ⊣

4.9 Definition (Colored term)

Let $P(s_1, \ldots, s_k)^n_m S$ be an atomic formula occurrence in a leaf sequent. For all s_i, where $1 \leq i \leq k$, the color S gives the corresponding *colored term* $s_i \oplus S$ like this:

- $u^n_m \oplus S = u^n_m S$
- $f^n_m(t_1, \ldots, t_l) \oplus S = f^n_m(t_1 \oplus S, \ldots, t_l \oplus S)$

Notice that function symbols are not colored in the same way as instantiation variables are colored; if a^n_m is a unary Skolem function, then $a^n_m \oplus S = a^n_m$. We will use tS as an abbreviation for the colored term $t \oplus S$ when the context allows it. All colored instantiation variables in a colored term are colored with the same color. ⊣

4.10 Definition (Connection)

A *connection* c in π is a subsequent $P(s_1, \ldots, s_k)S \vdash P(t_1, \ldots, t_k)T$ of a leaf sequent in π, where $P(s_1, \ldots, s_k)^{n_1}_{m_1}$ and $P(t_1, \ldots, t_k)^{n_2}_{m_2}$ are two atomic formulas with the same predicate symbol P. The associated *equation set* for c, written $\mathsf{Eq}(c)$, is the set of equations $\{s_i S = t_i T \mid 1 \leq i \leq k\}$. Each equation is between two colored terms. If C is a set of connections, then the equation set for C, written $\mathsf{Eq}(C)$, is the union of all equation sets for connections in C, i.e. $\mathsf{Eq}(C) = \bigcup \{\mathsf{Eq}(c) \mid c \in C\}$. A set C of connections is *spanning* for π if there is exactly one connection in C for each branch in π. ⊣

4.11 Definition (Herbrand universe, for connections)

Let C be a spanning set of connections and E be the associated equation set for C. The set of *colored variables* of C, written $\mathsf{Cvar}(C)$, is the set of all *colored* instantiation variables occurring in an equation in E. The *Herbrand universe* of C, written $\mathsf{Her}(C)$, is the set of all terms that can be generated from the function symbols occurring in C together with $\mathsf{Cvar}(C)$. If there are no constant symbols in C, a dummy constant is added to the Herbrand universe. ⊣

4.12 Definition (Substitution, for connections)

Let C be a spanning set of connections and E be the associated equation set for C. A *substitution* for C is a function σ from the colored variables of C to the Herbrand universe of C. It is extended to a substitution from the Herbrand universe of C to the Herbrand universe of C in the usual way. It is a *closing substitution* for C if it satisfies each equation $sS = tT$ in E, that is, if $(sS)\sigma = (tT)\sigma$. ⊣

We will from now on assume that all function symbols occurring in a skeleton are Skolem functions. This is not strictly necessary, but it will facilitate the exposition a good deal. For instance, if a function symbol occurs in a connection, then we will know for sure that this is a Skolem function and that there is a δ-inference somewhere in the branch which introduces it. Without this simplification, the definitions have more cases and extra care must be taken.

Example 4.13 A skeleton for the following *valid* sequent is given below.

$$\underset{1}{\forall} \underset{2}{x} Px \vdash \underset{3}{\forall x} \underset{4}{\forall y}(\underset{6}{Px} \underset{5}{\wedge} \underset{7}{Pz})$$

Let u abbreviate u_1^1, a abbreviate a_3^1 and b abbreviate a_4^1.

$$\cfrac{\cfrac{\cfrac{\cfrac{Pu\{{}^1_6\} \vdash Pa \quad Pu\{{}^1_7\} \vdash Pb}{Pu \vdash Pa \wedge Pb}}{\forall x Px \vdash Pa \wedge Pb}\, u}{\forall x Px \vdash \forall y (Pa \wedge Py)}\, b}{\forall x Px \vdash \forall x \forall y (Px \wedge Pz)}\, a$$

The set of leaf sequents is a spanning set of connections, and a closing substitution is $\{u\{{}^1_6\}/a, u\{{}^1_7\}/b\}$.

At this point, one might be inclined to suggest the following closure condition for LK^{s}: *"A skeleton π is closable if there exists a spanning set of connections C and a closing substitution for C."* But, this is not correct, even though it works for the example above. We shall soon see what goes wrong if appropriate care is not taken.

From this point and outwards this thesis should be considered a *work in progress*; what follows is a fairly detailed discussion of our current intuitions. A lot effort has been made to understand all the concepts involved, but the matter is fairly complex and a finished, complete result can unfortunately not be given at this point.

Example 4.14 The following sequent is not valid.

$$\underset{1}{\forall x}(\underset{3}{Px} \underset{2}{\vee} \underset{4}{Qx}), \underset{5}{\forall x}(\underset{7}{Px} \underset{6}{\vee} \underset{8}{Qx}) \vdash \underset{9}{\forall x} \underset{10}{Px}, \underset{11}{\forall x} \underset{12}{Qx}$$

But with the erroneous definition of a closure condition, it would be provable. Let v,w,b,c abbreviate u_1^1, u_5^1, a_9^1, a_{11}^1, respectively.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\mathbf{1}) \qquad (\mathbf{2})}{Pv, Pw \lor Qw\{\genfrac{}{}{0pt}{}{1}{3}\} \vdash Pb\{\genfrac{}{}{0pt}{}{1}{3}\}, Qc\{\genfrac{}{}{0pt}{}{1}{3}\}}\,\mathsf{L}\lor \quad \dfrac{(\mathbf{3}) \qquad (\mathbf{4})}{Qv, Pw \lor Qw\{\genfrac{}{}{0pt}{}{1}{4}\} \vdash Pb\{\genfrac{}{}{0pt}{}{1}{4}\}, Qc\{\genfrac{}{}{0pt}{}{1}{4}\}}\,\mathsf{L}\lor}{Pv \lor Qv, Pw \lor Qw \vdash Pb, Qc}\,\mathsf{L}\lor}{Pv \lor Qv, Pw \lor Qw \vdash Pb, \forall x Qx}\,c}{Pv \lor Qv, Pw \lor Qw \vdash \forall x Px, \forall x Qx}\,b}{Pv \lor Qv, \forall x(Px \lor Qx) \vdash \forall x Px, \forall x Qx}\,w}{\forall x(Px \lor Qx), \forall x(Px \lor Qx) \vdash \forall x Px, \forall x Qx}\,v$$

$(\mathbf{1})$	$\underline{Pv\{\genfrac{}{}{0pt}{}{1}{7}\}}, Pw\{\genfrac{}{}{0pt}{}{1}{3}\} \vdash \underline{Pb\{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{7}\}}, Qc\{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{7}\}$	$v\{\genfrac{}{}{0pt}{}{1}{7}\} \mapsto b\{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{7}\}$
$(\mathbf{2})$	$Pv\{\genfrac{}{}{0pt}{}{1}{8}\}, \underline{Qw\{\genfrac{}{}{0pt}{}{1}{3}\}} \vdash Pb\{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{8}\}, \underline{Qc\{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{8}\}}$	$w\{\genfrac{}{}{0pt}{}{1}{3}\} \mapsto c\{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{8}\}$
$(\mathbf{3})$	$Qv\{\genfrac{}{}{0pt}{}{1}{7}\}, \underline{Pw\{\genfrac{}{}{0pt}{}{1}{4}\}} \vdash \underline{Pb\{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{7}\}}, Qc\{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{7}\}$	$w\{\genfrac{}{}{0pt}{}{1}{4}\} \mapsto b\{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{7}\}$
$(\mathbf{4})$	$\underline{Qv\{\genfrac{}{}{0pt}{}{1}{8}\}}, Qw\{\genfrac{}{}{0pt}{}{1}{4}\} \vdash Pb\{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{8}\}, \underline{Qc\{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{8}\}}$	$v\{\genfrac{}{}{0pt}{}{1}{8}\} \mapsto c\{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{8}\}$

A spanning set of connections is indicated by the underlined formula occurrences. To the right is a *closing* substitution for this set of connections.

To give some intuitions for what goes wrong here, consider the equations for the connections in **(1)** and **(4)**:

$(\mathbf{1})$	$v\{\genfrac{}{}{0pt}{}{1}{7}\} = b\{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{7}\}$	Intersection of splitting sets: $\{\genfrac{}{}{0pt}{}{1}{7}\} \cap \{\genfrac{}{}{0pt}{}{1}{3}\,\genfrac{}{}{0pt}{}{1}{7}\} = \{\genfrac{}{}{0pt}{}{1}{7}\}$
$(\mathbf{4})$	$v\{\genfrac{}{}{0pt}{}{1}{8}\} = c\{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{8}\}$	Intersection of splitting sets: $\{\genfrac{}{}{0pt}{}{1}{8}\} \cap \{\genfrac{}{}{0pt}{}{1}{4}\,\genfrac{}{}{0pt}{}{1}{8}\} = \{\genfrac{}{}{0pt}{}{1}{8}\}$

In equation **(1)**, the index pair $\frac{1}{7}$ occurs in *both* splitting sets, and in equation **(4)** the same is the case for $\frac{1}{8}$. This is significant. Let us remove the index pairs in the intersection from the splitting sets:

$(\mathbf{1'})$	$v = b\{\genfrac{}{}{0pt}{}{1}{3}\}$	$(Pv \vdash Pb\{\genfrac{}{}{0pt}{}{1}{3}\})$
$(\mathbf{4'})$	$v = c\{\genfrac{}{}{0pt}{}{1}{4}\}$	$(Qv \vdash Qc\{\genfrac{}{}{0pt}{}{1}{4}\})$

These two equations cannot be solved simultaneously. Suppose that the two topmost $\mathsf{L}\lor$-inference in the skeleton were absent and that the leaf sequents were the two premisses of the bottommost $\mathsf{L}\lor$-inference. Then we would have the two connections written to the right of the equations $(\mathbf{1'})$ and $(\mathbf{4'})$. There is no closing substitution for this spanning set of connections either. The removal of the intersection of the splitting sets corresponds to this fact. A closing substitution should satisfy such a condition; if the substitution solves the equations $(\mathbf{1})$ and $(\mathbf{4})$, then there should also be a substitution which solves $(\mathbf{1'})$ and $(\mathbf{4'})$. This is an informal description of the notion of *homogeneity*.

4.15 Definition (Reduced connection)

Let c be the connection $P(s_1, \ldots, s_k)S \vdash P(t_1, \ldots, t_k)T$. Let $X = S \cap T$. The corresponding *reduced connection* is the sequent $P(s_1, \ldots, s_k)(S \setminus X) \vdash P(t_1, \ldots, t_k)(T \setminus X)$. For any set C of connections, the corresponding set of reduced connections of C is denoted $\mathsf{Red}(C)$. $\dashv$

Example 4.16 The reduced connections for the spanning set of connections given in Example 4.14 are these:

$$\begin{array}{lc}
(\mathbf{1'}) & Pv \vdash Pb\{^1_3\} \\
(\mathbf{2'}) & Qw \vdash Qc\{^1_8\} \\
(\mathbf{3'}) & Pw \vdash Pb\{^1_7\} \\
(\mathbf{4'}) & Qv \vdash Qc\{^1_8\}
\end{array}$$

For this set there is no closing substitution.

In addition to substitutions for sets of connections, we will now consider substitutions for the corresponding sets of reduced connections. If C is a spanning set of connections, a substitution for $\mathsf{Red}(C)$ will be a function from the colored variables of $\mathsf{Red}(C)$ to the Herbrand universe of $\mathsf{Red}(C)$. Naturally, there will be *fewer* colored variables of $\mathsf{Red}(C)$ than colored variables of C.

4.17 Definition (Color extension)

Let C be a spanning set of connections and $\mathsf{Red}(C)$ be the corresponding set of reduced connections. A substitution σ for $\mathsf{Red}(C)$ can be extended to a substitution σ' for C, called a *color extension* of σ, in the following way.

Let uS be a colored variable of $\mathsf{Red}(C)$ and uT be a colored variable of C such that $S \subseteq T$. Let $X = T \setminus S$, i.e. the index pairs in T which are not in S. Then, $\sigma'(uT)$ is defined to be the term $\sigma(uS) \uplus X$, which is the term $\sigma(uS)$ where all splitting sets Y have been replaced with $Y \cup X$, i.e. all the index pairs of X have been added to all the splitting sets. $\dashv$

The only case where $\sigma(uS)$ and $\sigma(uS) \uplus X$ in the above definition differ, is when $\sigma(uS)$ contains colored variables, since function symbols are not colored with splitting sets. If we assume σ to be ground, we would not have to do this.

4.18 Definition (Homogeneity)

Let C be a spanning set of connections for a skeleton π. A substitution σ is *homogeneous* for C if it is the color extension of a *closing* substitution for $\mathsf{Red}(C)$. $\dashv$

Example 4.19 The substitution which was closing for the set of connections in Example 4.14 is not homogeneous, since it is not the color extension of a closing substitution for the set of reduced connections. In this case the set of reduced connections does not have a closing substitution.

4.20 Definition (Idempotent)

A substitution σ is *idempotent* if for all u it is the case that $u\sigma = (u\sigma)\sigma$. $\dashv$

4.21 Definition (Closure condition for $\mathsf{LK^s}$)

Let C be a spanning set of connections for a skeleton π. Then, $\langle \pi, C, \sigma \rangle$ is a *proof* of the root sequent in π if σ is idempotent, homogeneous and closing. $\dashv$

4.22 Definition (Substitution ordering, for $\mathsf{LK^s}$)

Let $\langle \pi, C, \sigma \rangle$ be $\mathsf{LK^s}$-proof. Let τ be a closing substitution for $\mathsf{Red}(C)$ such that σ is a color extension of τ. Assume without loss of generality that τ is *minimal* in the sense that every binding in the support of σ is necessary in order to close all branches of π. (A binding is necessary for the closure if the result of taking the binding out of the support is that some branch is not closed. If τ is not minimal in this sense, then we can remove the bindings which are not necessary.) Then, $r_u \sqsupset r_f$ holds if it is the case that:

(1) r_u and r_f are two inferences in the *same branch* β of π

(2) r_u introduces the instantiation variable u

(3) r_f introduces the Skolem function f

(4) $(uU)\tau = f(\vec{t})$, for some sequence of terms $\vec{t}$ in the Herbrand universe of $\mathsf{Red}(C)$

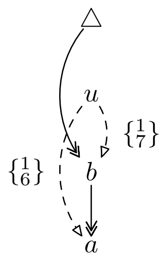
(5) u and f both occur in a connection in C which closes β

The relation $\sqsupset$ is the substitution ordering induced from σ. We say that $r_u \sqsupset r_f$ holds *with respect to* the splitting set U if $(uU)\tau = f(\vec{t})$ from point (4). $\dashv$

The diagram arrow between two inferences r_u and r_f, such that $r_u \sqsupset r_f$ will be drawn like usual, but with an extra label indicating with respect to which splitting set $r \sqsupset s$ holds.

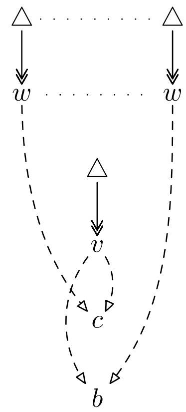
Example 4.23 A diagram represenation for the $\mathsf{LK^s}$-skeleton in Example 4.13. $u \sqsupset a$ holds with respect to $\{^1_6\}$ and $u \sqsupset b$ holds with respect to $\{^1_7\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{cc} u\{^1_6\}/a & u\{^1_7\}/b \\ Pu\{^1_6\} \vdash Pa & Pu\{^1_7\} \vdash Pb \end{array}}{Pu \vdash Pa \wedge Pb}}{\forall x Px \vdash Pa \wedge Pb}\, u}{\forall x Px \vdash \forall y(Pa \wedge Py)}\, b}{\forall x Px \vdash \forall x \forall y(Px \wedge Pz)}\, a$$



Example 4.24 The LK^{s}-skeleton in Example 4.14 does not have a closing and homogeneous substitution, but we can nevertheless consider a *hypothetical* closing substitution for the set of reduced connections. Below is the diagram representation of a permutation variant of the skeleton, where the diagram arrows for the hypothetical substitution is displayed.

$$\begin{array}{lrcl}
(\mathbf{1'}) & Pv & = & Pb\{^1_3\} \\
(\mathbf{2'}) & Qw & = & Qc\{^1_8\} \\
(\mathbf{3'}) & Pw & = & Pb\{^1_7\} \\
(\mathbf{4'}) & Qv & = & Qc\{^1_8\}
\end{array}$$



4.3 A BIG EXAMPLE

"Small examples are deceiving."

anonymous

The formula tree in Example 2.17 gives rise to the following valid sequent:

$$(\underset{3}{\forall x}\underset{4}{\exists y}(\underset{6}{Pxy} \underset{5}{\land} \underset{7}{Sy}))^1,\ (\underset{8}{\forall x}\underset{9}{\forall y}\underset{10}{\forall z}((\underset{13}{Pxy} \underset{12}{\land} \underset{14}{Pyz}) \underset{11}{\to} \underset{15}{Gxz}))^1 \vdash (\underset{16}{\forall x}\underset{17}{\exists y}(\underset{19}{Gxy} \underset{18}{\land} \underset{20}{Sy}))^1$$

Intuitively, the three formulas express *"Everyone has a Snill Parent"*, *"If y is the Parent of x and z is the Parent of y, then z is the Grandparent of x"* and *"x has a Snill Grandparent"*.

Let u abbreviate u^1_3, w abbreviate u^1_{17}, a abbreviate a^1_{16} and f abbreviate f^1_4.

$$
\dfrac{
 \dfrac{
  \dfrac{
   \dfrac{
    \dfrac{
     \dfrac{
      \dfrac{
       \dfrac{\dfrac{\mathbf{(1a)}\quad\mathbf{(1b)}}{\mathbf{(1')}}\,\mathsf{R}\land}{\mathbf{(1)}}\,w
       \qquad
       \dfrac{\dfrac{\mathbf{(2a)}\quad\mathbf{(2b)}}{\mathbf{(2')}}\,\mathsf{R}\land}{\mathbf{(2)}}\,w
      }{Pufu\{^1_{12}\}, Sfu\{^1_{12}\} \vdash \exists y(Gay \land Sy), Pu^1_8u^1_9 \land Pu^1_9u^1_{10}}\,\mathsf{R}\land
      \qquad
      \dfrac{\dfrac{\mathbf{(3a)}\quad\mathbf{(3b)}}{\mathbf{(3')}}\,\mathsf{R}\land}{\mathbf{(3)}}\,w
     }{Pufu, Sfu, (Pu^1_8u^1_9 \land Pu^1_9u^1_{10}) \to Gu^1_8u^1_{10} \vdash \exists y(Gay \land Sy)}\,\mathsf{L}\!\to
    }{Pufu, Sfu, \forall x\forall y\forall z((Pxy \land Pyz) \to Gxz) \vdash \exists y(Gay \land Sy)}\,u^1_8, u^1_9, u^1_{10}
   }{Pufu \land Sfu, \forall x\forall y\forall z((Pxy \land Pyz) \to Gxz) \vdash \exists y(Gay \land Sy)}\,\mathsf{L}\land
  }{\exists y(Puy \land Sy), \forall x\forall y\forall z((Pxy \land Pyz) \to Gxz) \vdash \exists y(Gay \land Sy)}\,fu
 }{(\forall x\exists y(Pxy \land Sy), \forall x\forall y\forall z((Pxy \land Pyz) \to Gxz)) \vdash \exists y(Gay \land Sy)}\,u
}{(\forall x\exists y(Pxy \land Sy), \forall x\forall y\forall z((Pxy \land Pyz) \to Gxz)) \vdash \forall x\exists y(Gxy \land Sy)}\,a
$$

(1a) $\underline{Pufu\{^1_{12}\,^1_{13}\,^1_{19}\}}, Sfu\{^1_{12}\,^1_{13}\,^1_{19}\} \vdash Gaw\{^1_{12}\,^1_{13}\}, \underline{Pu^1_8u^1_9\{^1_{19}\}}$

(1b) $\underline{Pufu\{^1_{12}\,^1_{13}\,^1_{20}\}}, Sfu\{^1_{12}\,^1_{13}\,^1_{20}\} \vdash Sw\{^1_{12}\,^1_{13}\}, \underline{Pu^1_8u^1_9\{^1_{20}\}}$

(2a) $\underline{Pufu\{^1_{12}\,^1_{14}\,^1_{19}\}}, Sfu\{^1_{12}\,^1_{14}\,^1_{19}\} \vdash Gaw\{^1_{12}\,^1_{14}\}, \underline{Pu^1_9u^1_{10}\{^1_{19}\}}$

(2b) $\underline{Pufu\{^1_{12}\,^1_{14}\,^1_{20}\}}, Sfu\{^1_{12}\,^1_{14}\,^1_{20}\} \vdash Sw\{^1_{12}\,^1_{14}\}, \underline{Pu^1_9u^1_{10}\{^1_{20}\}}$

(3a) $\underline{Gu^1_8u^1_{10}\{^1_{19}\}}, Pufu\{^1_{15}\,^1_{19}\}, Sfu\{^1_{15}\,^1_{19}\} \vdash \underline{Gaw\{^1_{15}\}}$

(3b) $Gu^1_8u^1_{10}\{^1_{20}\}, Pufu\{^1_{15}\,^1_{20}\}, \underline{Sfu\{^1_{15}\,^1_{20}\}} \vdash \underline{Sw\{^1_{15}\}}$

Figure 4.2: The six leaf sequents of the skeleton. A spanning set of connections C is indicated by the underlined formula occurrences.

$\mathsf{Red}(\mathbf{1a})$	$Pufu\{^{1}_{12}\ ^{1}_{13}\} \vdash Pu^{1}_{8}u^{1}_{9}$
$\mathsf{Red}(\mathbf{1b})$	$Pufu\{^{1}_{12}\ ^{1}_{13}\} \vdash Pu^{1}_{8}u^{1}_{9}$
$\mathsf{Red}(\mathbf{2a})$	$Pufu\{^{1}_{12}\ ^{1}_{14}\} \vdash Pu^{1}_{9}u^{1}_{10}$
$\mathsf{Red}(\mathbf{2b})$	$Pufu\{^{1}_{12}\ ^{1}_{14}\} \vdash Pu^{1}_{9}u^{1}_{10}$
$\mathsf{Red}(\mathbf{3a})$	$Gu^{1}_{8}u^{1}_{10}\{^{1}_{19}\} \vdash Gaw\{^{1}_{15}\}$
$\mathsf{Red}(\mathbf{3b})$	$Sfu\{^{1}_{20}\} \vdash Sw$

Figure 4.3: The set of reduced connections, $\mathsf{Red}(C)$.

σ'	relevant connections	color extension σ	
$u\{^{1}_{12}\ ^{1}_{13}\}/a$	$\mathbf{1a}, \mathbf{1b}$	$u\{^{1}_{12}\ ^{1}_{13}\ ^{1}_{19}\}/a$	$u\{^{1}_{12}\ ^{1}_{13}\ ^{1}_{20}\}/a$
u^{1}_{8}/a	$\mathbf{1a}, \mathbf{1b}$	$u^{1}_{8}\{^{1}_{19}\}/a$	$u^{1}_{8}\{^{1}_{20}\}/a$
u^{1}_{9}/fa	$\mathbf{1a}, \mathbf{1b}, \mathbf{2a}, \mathbf{2b}$	$u^{1}_{9}\{^{1}_{19}\}/fa$	$u^{1}_{9}\{^{1}_{20}\}/fa$
$u\{^{1}_{12}\ ^{1}_{14}\}/fa$	$\mathbf{2a}, \mathbf{2b}$	$u\{^{1}_{12}\ ^{1}_{14}\ ^{1}_{19}\}/fa$	$u\{^{1}_{12}\ ^{1}_{14}\ ^{1}_{20}\}/fa$
u^{1}_{10}/ffa	$\mathbf{2a}, \mathbf{2b}$	$u^{1}_{10}\{^{1}_{19}\}/ffa$	$u^{1}_{10}\{^{1}_{20}\}/ffa$
$u^{1}_{10}\{^{1}_{19}\}/ffa$	$\mathbf{3a}$	$u^{1}_{10}\{^{1}_{19}\}/ffa$	
$u^{1}_{8}\{^{1}_{19}\}/a$	$\mathbf{3a}$	$u^{1}_{8}\{^{1}_{19}\}/a$	
$w\{^{1}_{15}\}/ffa$	$\mathbf{3a}$	$w\{^{1}_{15}\}/ffa$	
w/ffa	$\mathbf{3b}$	$w\{^{1}_{15}\}/ffa$	
$u\{^{1}_{20}\}/fa$	$\mathbf{3b}$	$u\{^{1}_{15}\ ^{1}_{20}\}/fa$	

Figure 4.4: A closing substitution σ' for $\mathsf{Red}(C)$, together with a color extension σ which is closing for C.

In Figure 4.2, the six leaf nodes of the skeleton are displayed, and a spanning set C of connections is indicated by the underlined formula occurrences. A proof of the root sequent is $\langle \pi, C, \sigma \rangle$, where σ is the closing substitution given in the rightmost column of Figure 4.4. This substitution is homogeneous, since it is the color extension of the substitution σ' given in the leftmost column of the same figure. The substitution σ' is closing for the set of reduced connections. The middle column of the figure indicates which connections that are closed by the bindings of σ and σ', respectively. In Figure 4.5 there is a diagram representation.

It should be noted that C is not the only possible spanning set of connections. The leaf sequents (**1b**) and (**2b**) could instead give rise to the connections:

$$(\mathbf{1b}') \qquad Sfu\{^1_{12}\ ^1_{13}\ ^1_{20}\} \vdash Sw\{^1_{12}\ ^1_{13}\}$$

$$(\mathbf{2b}') \qquad Sfu\{^1_{12}\ ^1_{14}\ ^1_{20}\} \vdash Sw\{^1_{12}\ ^1_{14}\}$$

The corresponding reduced connections, both *identical* to $\mathsf{Red}(\mathbf{3b})$, are:

$$\mathsf{Red}(\mathbf{1b}') \qquad Sfu\{^1_{20}\} \vdash Sw$$

$$\mathsf{Red}(\mathbf{2b}') \qquad Sfu\{^1_{20}\} \vdash Sw$$

The same substitution σ', for the same set of reduced connections, has a color extension which is closing for the new set of connections. This color extension, together with the skeleton and the new set of connections, gives another proof of the root sequent.

4.3.1 Remark on non-balanced skeletons

Consider the skeleton which has (**1**), (**2**), (**3a**) and (**3b**) as leaf sequents:

$$(\mathbf{1}) \quad \underline{Pufu\{^1_{12}\ ^1_{13}\}}, Sfu\{^1_{12}\ ^1_{13}\} \vdash \exists y(Gay \land Sy)\{^1_{12}\ ^1_{13}\}, \underline{Pu^1_8 u^1_9}$$

$$(\mathbf{2}) \quad \underline{Pufu\{^1_{12}\ ^1_{14}\}}, Sfu\{^1_{12}\ ^1_{14}\} \vdash \exists y(Gay \land Sy)\{^1_{12}\ ^1_{14}\}, \underline{Pu^1_9 u^1_{10}}$$

$$(\mathbf{3a}) \quad \underline{Gu^1_8 u^1_{10}\{^1_{19}\}}, Pufu\{^1_{15}\ ^1_{19}\}, Sfu\{^1_{15}\ ^1_{19}\} \vdash \underline{Gaw\{^1_{15}\}}$$

$$(\mathbf{3b}) \quad Gu^1_8 u^1_{10}\{^1_{20}\}, Pufu\{^1_{15}\ ^1_{20}\}, \underline{Sfu\{^1_{15}\ ^1_{20}\}} \vdash \underline{Sw\{^1_{15}\}}$$

This skeleton is not balanced. Even though is is not yet clear what the appropriate conditions for a closing substitution should be for non-balanced skeletons, it is interesting to see how far the concepts from the balanced case can take us.

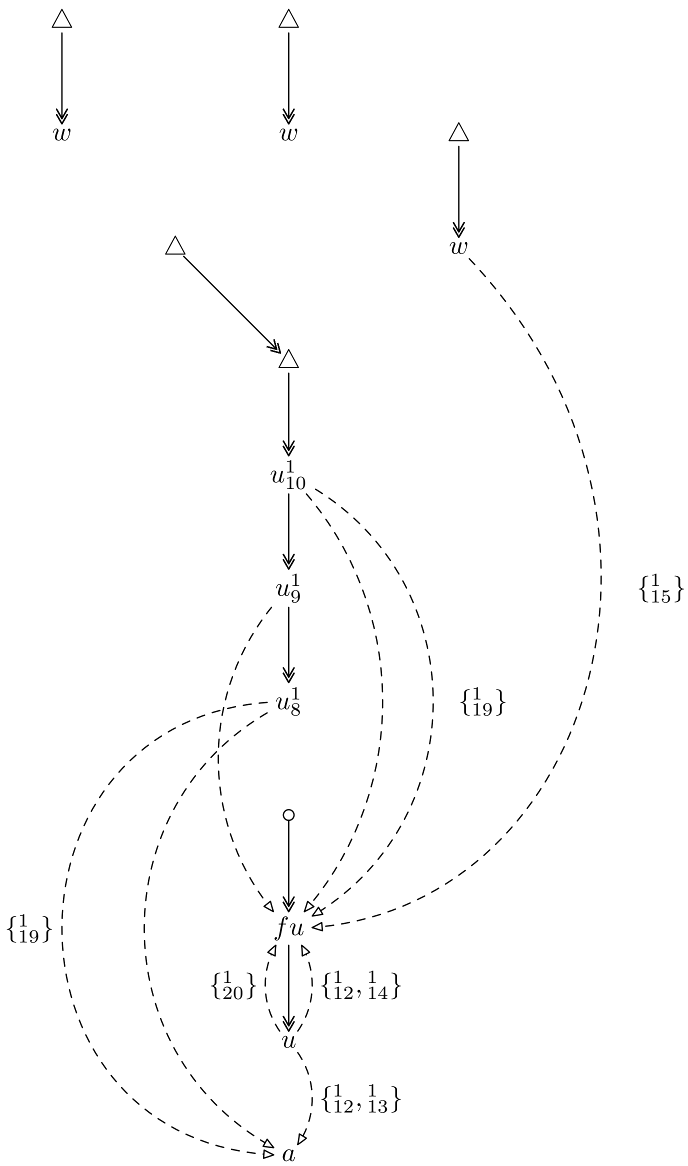


Figure 4.5: A diagram representation. (Not showing arrows for $w \gg a$.)

The four connections which are possible are the subsequents which are underlined. This set of connections is spanning. The only connection which is not itself reduced is (**3b**). The reduced connection corresponding to (**3b**) is:

$\mathsf{Red}(\mathbf{3b}) \qquad Sfu\{^{1}_{20}\} \vdash Sw$

The substitution σ' in Figure 4.4 is closing for this reduced set of connections. Moreover, it is closing for the four underlined connections. So if we view σ' as its own color extension, we have a proof, even though the skeleton is not balanced.

One intuition for allowing non-balanced skeletons, like this one, is that there are corresponding proofs in LK, or equivalently LK^{δ}, which are not balanced.

Remark. This is not to be taken as a complete exposition of the non-balanced case, not even as a conjecture, but as an indication that a good characterization of provability for the non-balanced case is possible.

4.4 SOUNDNESS

We now turn to the proof of soundness for LK^{s}; that every provable sequent is valid. The core idea is to perform proof transformations on LK^{s}-proofs in order to obtain proofs which essentially are like proofs in LK or LK^{δ}. By examining relations between inferences in skeletons it is possible to give a precise characterization of how LK^{s}-proofs differ from LK-proofs, and this information enables us perform the correct transformation steps. There are three important properties which we want LK^{s}-proofs to satisfy in order to be LK-like. (1) The reduction ordering should be cycle-free, (2) the skeleton should be conforming and (3) the substitution ordering should be projective (defined below).

4.25 Definition (Conforming)

Let $\langle \pi, C, \sigma \rangle$ be a an LK^{s}-proof. The skeleton π *conforms to* the induced substitution ordering $\sqsupset$ if for all inferences r and s in π, such that $r \sqsupset s$, it is the case that r is above s. ⊣

4.26 Definition (Reduction ordering)

The induced reduction ordering $\rhd$ is defined exactly as for $\mathsf{LK}^{\mathsf{ce}}$. ⊣

With the induced reduction ordering we can speak of cycles for LK^{s}-proofs.

Example 4.27 The induced reduction ordering from the big example in Section 4.3 contains two different cycles. (1) $u \sqsupset f$ (wrt. $\{^{1}_{20}\}$) and $f \gtrsim u$.

(2) $u \sqsupset f$ (wrt. $\{^1_{12}, ^1_{14}\}$) and $f \gtrsim u$. This is easily seen from the diagram representation in Figure 4.5.

4.28 Definition (Projective)

Let $\langle \pi, C, \sigma \rangle$ be a an LK^{s}-proof. The induced substitution ordering $\sqsupset$ is *projective* if $r \sqsupset s_1$ and $r \sqsupset s_2$ implies that $s_1 = s_2$. If the $\sqsupset$ is projective, we say that the proof is projective. $\dashv$

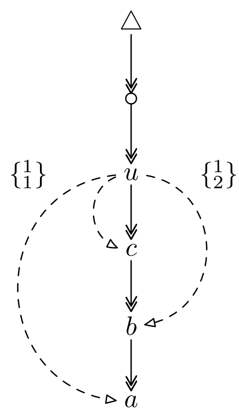
Example 4.29 The diagram in Example 4.23 shows that the induced substitution ordering is not projective, since $u \sqsupset a$ (wrt. $\{^1_6\}$) and $u \sqsupset b$ (wrt. $\{^1_6\}$), but not $a = b$. The substitution ordering from the big example in Section 4.3 not projective either, since $u \sqsupset a$ and $u \sqsupset f$.

4.30 Definition (LK-like)

Let $\langle \pi, C, \sigma \rangle$ be an LK^{s}-proof, and let $\rhd$ and $\sqsupset$ be the induced reduction and substitution ordering, respectively. The proof is LK-*like* if (1) $\rhd$ is acyclic, (2) π conforms to $\sqsupset$ and (3) $\sqsupset$ is projective. $\dashv$

Example 4.31 Let a,b,c and u be abbreviations for appropriate Skolem functions and instantiation variables. (The extra copy of the γ-formula is not displayed in the skeleton.) A proof in LK^{s}-of the sequent is given below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Pau \vdash Puc\{^1_1\} \qquad Pbu \vdash Puc\{^1_2\}}{Pau \lor Pbu \vdash Puc}\,\mathsf{L}\lor}{\vdash Pau \lor Pbu \to Puc}\,\mathsf{R}\!\to}{\vdash \exists u(Pau \lor Pbu \to Puc)}\,u}{\vdash \forall z \exists u(Pau \lor Pbu \to Puz)}\,c}{\vdash \forall y \forall z \exists u(Pau \lor Pyu \to Puz)}\,b}{\vdash \forall x \forall y \forall z \exists u(Pxu \lor Pyu \to Puz)}\,a$$



Both leaf sequents display reduced connections (the set of connections is identical to the set of reduced connections in this case). The equation set is $\{a = u\{^1_1\}, b = u\{^1_2\}, u = c\}$. A closing, homogeneous substitution is indicated above the leaf sequents. The substitution ordering is not projective, since $u \sqsupset a$ and $u \sqsupset b$, but not $a = b$. The substitution ordering is conforming.

The above example is interesting, because the variable u plays the role of *both* a rigid variable (u *without* splitting set occurs in both branches) and a universal variable (u occurs with *different* splitting sets in both branches).

4.32 Conjecture (Soundness of LK^{s})

A proof $\langle \pi, C, \sigma \rangle$ can be extended to a proof which is LK-like. $\dashv$

The proof of this result, which yields the consistency of the system, is currently not known. The idea for the proof is to threefold: (1) To show that every proof has a cycle-free extension (like for $\mathsf{LK}^{\mathsf{ce}}$). (2) Find a permutation variant of the proof which is conforming. (3) To show that every conforming proof has a projective extension. We formulate these three steps below and give a sketch of how this can be achieved.

4.33 Lemma (Cycle elimination) For every proof $\langle \pi, C, \sigma \rangle$ there is an extension of π with a idempotent, homogeneous and closing substitution σ' such that the induced reduction ordering $\triangleright$ is cycle-free. $\dashv$

The idea is still to "*break up*" cycles like we described for $\mathsf{LK}^{\mathsf{ce}}$. (See the article in the Appendix for one approach.)

4.34 Lemma (Conformity) For every proof $\langle \pi, C, \sigma \rangle$ in LK^{s} such that the reduction ordering $\triangleright$ is cycle-free, there is a conforming permutation variant. $\dashv$

Proof. Like for Lemma 3.22 for $\mathsf{LK}^{\mathsf{ce}}$. The reduction ordering is well-founded and a conforming permutation variant can be constructed by repeatedly choosing $\triangleright$-minimal inferences and permuting the skeleton accordingly. $\square$

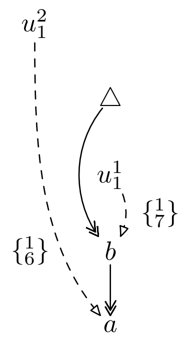
4.35 Lemma (Projection) Every cycle-free and conforming proof has a projective extension. $\dashv$

By repeatedly adding γ-inferences to the skeleton, removing the bindings from the substitution which makes the induced substitution ordering non-projective and introducing new bindings (keeping the substitution idempotent, homogeneous and closing), it should be possible to eliminate all "non-projective" parts of a closing substitution. (See the examples below.)

Example 4.36 Let us reconsider the proof from Example 4.13. It is cycle-free and conforming, but not projective. A projective extension is given below.

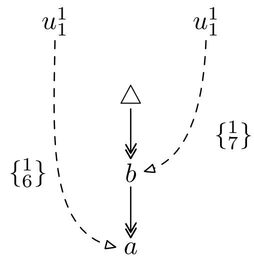
$$\underset{1}{\forall} x \underset{2}{P} x \vdash \underset{3}{\forall} x \underset{4}{\forall} y (\underset{6}{P} x \underset{5}{\wedge} \underset{7}{P} z)$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c} u_1^2\{\binom{1}{6}\}/a \\ \dfrac{Pu_1^2\{\binom{1}{6}\}, Pu_1^1\{\binom{1}{6}\} \vdash Pa}{(\forall x Px)\{\binom{1}{6}\}, Pu_1^1\{\binom{1}{6}\} \vdash Pa} \end{array} \quad \begin{array}{c} u_1^1\{\binom{1}{7}\}/b \\ (\forall x Px)\{\binom{1}{7}\}, Pu_1^1\{\binom{1}{7}\} \vdash Pb \end{array}}{\forall x Px, Pu_1^1 \vdash Pa \wedge Pb}}{\forall x Px \vdash Pa \wedge Pb}\, u_1^1}{\forall x Px \vdash \forall y (Pa \wedge Py)}\, b}{\forall x Px \vdash \forall x \forall y (Px \wedge Pz)}\, a
$$

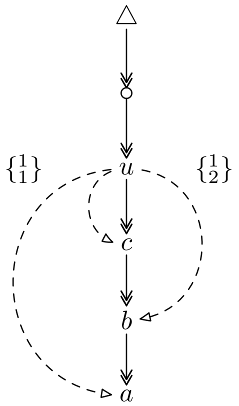


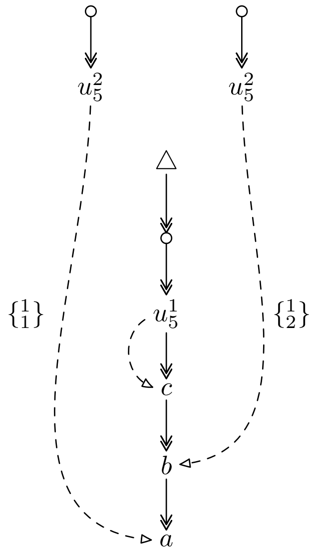
It is interesting to see that it is not *necessary* to introduce another γ-inference; in this case there is a projective permutation variant, due to the fact that the β-inference can occur below the γ-inference in the skeleton.

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c} u_1^1\{\binom{1}{6}\}/a \\ \dfrac{Pu_1^1\{\binom{1}{6}\} \vdash Pa}{(\forall x Px)\{\binom{1}{6}\} \vdash Pa}\, u_1^1 \end{array} \quad \begin{array}{c} u_1^1\{\binom{1}{7}\}/b \\ \dfrac{Pu_1^1\{\binom{1}{7}\} \vdash Pb}{(\forall x Px)\{\binom{1}{7}\} \vdash Pb}\, u_1^1 \end{array}}{\forall x Px \vdash Pa \wedge Pb}}{\forall x Px \vdash \forall y (Pa \wedge Py)}\, b}{\forall x Px \vdash \forall x \forall y (Px \wedge Pz)}\, a
$$



Example 4.37 The proof from Example 4.31 can be made projective by expanding the γ-formula, $\exists u(Pau \vee Pbu \rightarrow Puc)$ in both branches; thus removing both of the bindings $u\{\binom{1}{1}\}/a$ and $u\{\binom{1}{2}\}/b$. It is not necessary to expand the β-subformula, because a closing, homogeneous substitution is reached already after applying $\mathsf{R}\rightarrow$. In the diagram below, we have assumed that u was an abbreviation for u_5^1.





Instead of giving the non-projective $\mathsf{LK^s}$-proof, let us see that this LK-like proof actually gives a proof in LK!

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\boxed{Pac}, Paa \lor Pba \vdash Pcc, \boxed{Pac}}{Pac \vdash Pcc, Paa \lor Pba \to Pac}}{Pac \vdash Pcc, \exists u(Pau \lor Pbu \to Puc)}\,\gamma_a \quad \dfrac{\dfrac{\boxed{Pbc}, Pab \lor Pbb \vdash Pcc, \boxed{Pbc}}{Pbc \vdash Pcc, Pab \lor Pbb \to Pbc}}{Pbc \vdash Pcc, \exists u(Pau \lor Pbu \to Puc)}\,\gamma_b}{Pac \lor Pbc \vdash Pcc, \exists u(Pau \lor Pbu \to Puc)}\,\mathsf{L}\lor}{\vdash Pac \lor Pbc \to Pcc, \exists u(Pau \lor Pbu \to Puc)}\,\mathsf{R}\!\to}{\vdash \exists u(Pau \lor Pbu \to Puc)}\,\gamma_c}{\vdash \forall z \exists u(Pau \lor Pbu \to Puz)}\,c}{\vdash \forall y \forall z \exists u(Pau \lor Pyu \to Puz)}\,b}{\vdash \forall x \forall y \forall z \exists u(Pxu \lor Pyu \to Puz)}\,a
$$

The diagram representation of the LK-like proof corresponds *exactly* to the actual LK-proof. All the essential relations between the LK-inferences are captured in this diagram.

4.5 A FRAMEWORK FOR REASONING ABOUT SPLITTING

The introduction of splitting sets and index pairs enables a framework in which it is possible to reason very explicitly about branches in a skeleton and the formula occurrences in these branches. Since β-formulas split branches and add index pairs to the surrounding formula occurrences, there is information available to us which was not available before. This section introduces some new terminology and concepts in order to speak of about splitting in a precise and constructive way.

In all the definitions below, we assume that a finite set of formula trees (see Section 2.4) is given, and that all index pairs of these are distinct. All the examples refer to the formula tree in Figure 4.5, and only the occurrence numbers of the index pairs will be given.

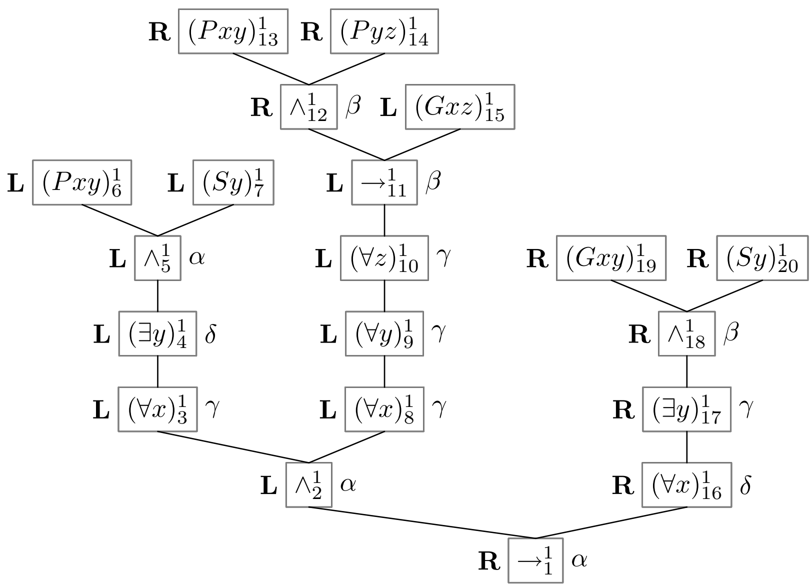


Figure 4.6: The formula tree from Example 2.17.

4.38 Definition (β-related nodes)

Two different nodes x and y in a formula tree are *β-related*, written $x \parallel_\beta y$, if (1) they are not in the same branch of the formula tree, and (2) their greatest common descendant in the formula tree is of principal type β. $\dashv$

Example 4.39 The following nodes are β-related: $12\|_\beta 15$ (gcd is 11), $13\|_\beta 15$ (gcd is 11), $14\|_\beta 15$ (gcd is 11), $13\|_\beta 14$ (gcd is 12) and $19\|_\beta 20$ (gcd is 18).

4.40 Definition (β-option, duality)

If ${n \atop m}$ is a β-node, then the index pairs of the immediate ancestors are called the *β-options* for ${n \atop m}$; the two index pairs are called *dual*. The β-options for ${n \atop m}$ are given by $\beta_1\binom{n}{m}$ and $\beta_2\binom{n}{m}$. If $\beta_i\binom{n}{m}$ equals ${l \atop k}$, then the dual is given by $\overline{\beta_i\binom{n}{m}}$ or $\frac{l}{k}$. If S is set of index pairs, then $\overline{S} = \{\frac{n}{m} \mid {n \atop m} \in S\}$. $\dashv$

Example 4.41 The set of β-options for the formula tree is $\{12, 13, 14, 15, 19, 20\}$. The pairs $\langle 13, 14\rangle$, $\langle 12, 15\rangle$ and $\langle 19, 20\rangle$ are dual index pairs.

4.42 Definition (β-successor)

Let ${n \atop m}$ be a node (not necessarily a β-node). A *β-successor* of ${n \atop m}$ is an ancestor of ${n \atop m}$ such that (1) it is a β-option and (2) there are no other β-options between it and ${n \atop m}$ in the formula tree. $\dashv$

Example 4.43 Both 12 and 15 are β-successors of 9, but 13 is *not* a β-successor of 9, since 12 is between them.

4.44 Definition (Complete skeleton)

A skeleton is *complete* if the following conditions hold: (1) If $\varphi^n_m S$ is a formula occurrence of principal type α, β or δ in the skeleton, then φ has an ancestor formula occurrence in the skeleton. (2) If $\varphi^n_m S$ is a formula occurrence of principal type γ in the skeleton, then there is either a formula occurrence $\varphi^k_m S$ in the skeleton, where $k < n$, or $\varphi^n_m S$ has an ancestor in the skeleton. $\dashv$

4.45 Definition (β-closed)

A set of index pairs b is *β-closed* if for all ${n \atop m}$ in b: if ${n' \atop m'}$ is a β-successor of ${n \atop m}$, then either ${n' \atop m'}$ or its dual is in b. (This condition corresponds to expanding a formula maximally in a derivation, which is what is required for complete skeletons.) All sets of index pairs can be extended to a β-closed set by repeatedly choosing β-successors for index pairs in the set. This process will be referred to as closing the set upwards under β-successors or performing a *β-closure*. A *partial β-closure* consists of adding only some β-successors in the desribed way. $\dashv$

4.46 Definition (β-chain)

Let ${n \atop m}$ be a node. A *β-chain* rooted in ${n \atop m}$ is a partial β-closure of $\{{n \atop m}\}$. $\dashv$

4.47 Definition (β-consistent)

A set of index pairs b is *β-consistent* if no two index pairs in b are β-related. $\dashv$

Trivially, dual index pairs can not both be in a β-consistent set, but that a set is β-consistent says more than this. If an index pair ${n \atop m}$ is in the set, then no index pair $\frac{n'}{m'}$ such that $\frac{n}{m}$ and $\frac{n'}{m'}$ have a β-node as the greatest common descendant, is allowed. The intuition behind this is that $\frac{n}{m}$ and $\frac{n'}{m'}$ should not be ancestors of dual β-options. A β-option excludes its dual and all ancestors of its dual, even though it is not itself in the β-consistent set.

4.48 Definition (β-path)

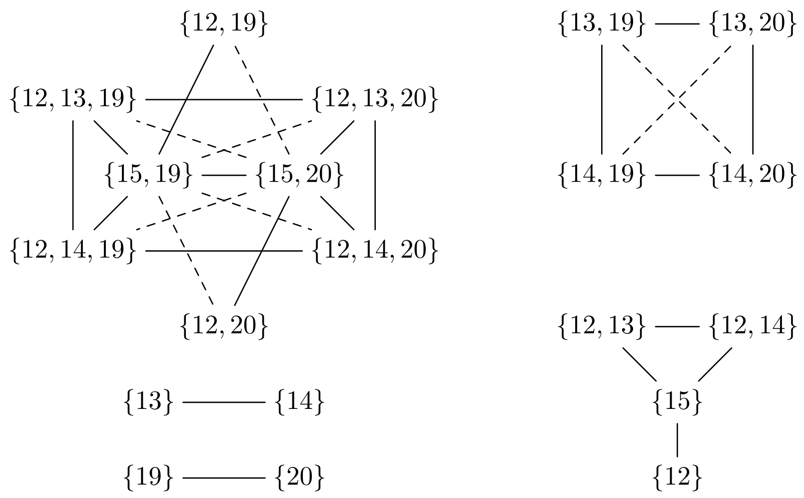
A *β-path* in a formula tree is a β-consistent set of β-options from the formula tree. A *complete β-path* is a β-closed β-path. $\dashv$

4.49 Definition (Complementary β-paths)

Two β-paths are *complementary* if one can be obtained from the other by performing finitely many β-changes. A β-change consists of (1) removing one index pair $\frac{n}{m}$ together with all its ancestors and (2) adding a β-chain rooted in the dual β-option $\frac{n}{m}$. $\dashv$

Notice, for instance, that each singleton set consisting of a β-option is trivially a β-path. If this β-option has no β-ancestors, then it is a complete β-path. The empty set is always a β-path.

Example 4.50 All β-paths are displayed below, where complementary β-paths are indicated with lines between them. The reader should verify that following a solid line consists of performing exactly one β-change, while following a dashed line consists of performing two β-changes, as explained in the definition.



For instance, to see that $\{13, 19\}$ is a β-path, observe that it is β-closed, since there are no additional ancestor β-options to add, and that it is β-consistent, since the greatest common descendant of 13 and 19 is the root node 1, which is *not* a β-node. The β-paths $\{13, 19\}$ and $\{14, 20\}$ are complementary.

One β-change consists of removing 13 and adding 14; another consists of removing 19 and adding 20.

Furthermore, the sets $\{13, 15\}$ and $\{14, 15\}$ are not β-paths, since they are *not β-consistent.* The greatest common descendant for each set is 11, which is a β-node. The sets $\{12, 19\}$ and $\{12, 20\}$ are β-paths, but not complete, since they are *not β-closed.* 13 is an ancestor β-option of 12, but it is not in any of the sets.

4.51 Lemma (β-path) Let $(P\vec{s})S \vdash (P\vec{t})T$ be a connection for a complete skeleton π. Then, $S \cap T$ is a β-path. $\dashv$

Proof. (1) $S \cap T$ is β-closed. Let $\frac{n}{m}$ be an index pair in $S \cap T$. Then, neither of the atomic formulas in the connection are ancestors of the formula with index pair $\frac{n}{m}$. If they were, $\frac{n}{m}$ would not be in their splitting sets. Therefore, neither of the atomic formulas in the connection are ancestors of any β-successors of $\frac{n}{m}$. Furthermore, if $\frac{n'}{m'}$ is a β-successor of $\frac{n}{m}$, then, since π is complete, the index pair $\frac{n'}{m'}$ is added to the splitting sets of all extra formula occurrences in one of the premisses; the dual index pair is added to the splitting sets of all extra formula occurrences in the other premiss. One of these index pairs must be added to both of the splitting lists of the atomic formulas, or descendants of these, so one of these index pairs must be in both S and T, and thus in $S \cap T$. (2) $S \cap T$ is β-consistent. By virtue of occurring in the leaf node of a branch in the skeleton, there are no two index pairs in $S \cap T$ which are β-related. $\square$

4.6 Summary and remarks

This is not the last word on either cycle elimination or uniform variable splitting; there is work left to be done. More precisely, here are the shortcomings:

- Cycle elimination for $\mathsf{LK}^{\mathsf{ce}}$ (Conjecture 3.25) and LK^{s}.
- Soundness of LK^{s} (Conjecture 4.32) needs to be properly established. It is believed that the this will follow from the proof of cycle elimination.
- Minimal closure conditions for LK^{s} needs to be found; what are the *minimal requirements* of a closure condition for LK^{s}-skeletons in order to obtain a sound calculus? The present approach can be found in Definition 4.18 of homogeneity. It is rather fresh and has probably not reached its final form. (See comments on article below for more details.)

The contributions of this thesis are briefly the following:

- A way of characterizing LK-likeness by means of intuitive diagrams representing relations between inferences. This gives a perspective on *proof transformation* which is new. Also, the problem of cycle elimination has been identified; cycles in a reduction ordering must be eliminated in order to obtain an LK-like proof.
- A new free variable calculus which utilizes splitting of variables in a systematic way. This is done by syntactically identifying variable dependencies across branches in a skeleton. Since variables can occur both as rigid and universal in a skeleton, this joins the two notions into a more general one.

After submission of the article [45], we have found some errors and room for improvement.

- The definition of homogeneity in the article differs from the one in this thesis because the one given in the article is too strong. In the article, there is the requirement that a certain colored instantiation variable, where one index pair has been replaced with a dual one, is in the domain of the substitution. This is not necessary, and the discovery of this gave birth to Example 4.50.
- Quote: *"To compensate for this* [splitting of variables] *the logical dependencies of some variables will be regained by introducing term equations which identify certain instantiation variables."* It is the author belief that the introduction of these auxiliary equation sets (denoted $\Pi(c)$ in the article) is not necessary. This was a technical step in the article to ensure that the substitution orderings were *projective*. Instead of doing this, it should be possible to extend a skeleton, like explained in Section 4.4, in order to make a substitution ordering projective.
- The Lemma *Existence of complementary colors* is not correct; there are simple counterexamples.

4.6.1 IDEAS FOR FURTHER DEVELOPMENT

Working with this new machinery of variable splitting, many new problems and topic arise. Here is a sketch of some of them.

Proof length How does LK^{s} relate to other free variable systems with respect to proof complexity and proof length? Baaz and Fermüller [4] has shown that the use of the δ^{++}-rule instead of the δ- and the δ^{+}-rule (and the use of the δ^{*}-rule instead of the δ^{++}-rule) gives non-elementary shortenings of proofs in the worst case. (They relate proof complexity of different calculi to the calculus-independent notion of *Herbrand* *complexity*.) A relevant question is therefore: Does LK^s allow for similar speed-up results?

Other δ-rules Even though LK^s is defined with a δ^{ce}-rule, it is also possible to use the other variants of δ-rule. (For instance, a δ-formula with index pair ${}^n_m$ could introduce f_m instead of f^n_m, which is closer to the δ^{++}-rule.) What happens if the idea of uniform variable splitting is applied to calculi with δ^{++}-, δ^*-, δ^{*^*}- and δ^ϵ-rules, respectively?

Structural rules and cut Weakening and contraction are not defined for LK^s. The splitting sets are developed *from below* by generating a skeleton from a given root sequent. It would be interesting to develop, if possible, a *full calculus* with structural rules like weakening and contraction, axioms and a cut rule. Then, some sort of *merging* of colors must be defined, which makes it possible to go from premisses to conclusions in a synthetical way. Additionally, it would be nice that have a cut-elimination proof for LK^s.

Cut introduction Investigate how cycle elimination can be done by means of introducing cuts.

Algebra Splitting sets and β-paths have an intrinsic structure which probably can be captured by means of algebraic tools. (Section 4.5 - A framework for reasoning about splitting - is the beginning of such an approach.)

The non-balanced case Investigate the conditions under which non-balanced skeletons of LK^s are permissible

The incremental closure technique This has been developed by Giese [23] and is a way of organizing the search for a closing substitution in an efficient way. It would be interesting to develop such an algorithm for LK^s as well. As a side product, we could show completeness of LK^s directly, by specifying a fair strategy.

Other logics Apply the method of variable splitting to labeled systems of modal and intuitionistic logic [44] and linear logic.

Permutations To characterize a permutation variant as a "merging" of formula trees. A skeleton can be viewed as a disjoint union of $\gg$-trees, and each $\gg$-tree, up to contextual equivalence, is isomorphic to a formula tree. Thus, a permutation variant of a skeleton corresponds to one way of organizing the formula trees in a skeleton.

Appendix A

Article

A free variable sequent calculus with uniform variable splitting

Arild Waaler[1] and Roger Antonsen[2]

[1] Finnmark College, 9500 Alta and
University of Oslo, Department of Informatics, 0315 Oslo, Norway.
arild@ifi.uio.no
[2] University of Oslo, Norway
rantonse@ulrik.uio.no

Abstract. A system with variable splitting is introduced for a sequent calculus with free variables and run-time Skolemization. Derivations in the system are invariant under permutation, so that the order in which rules are applied has no effect on the leaves. Technically this is achieved by means of a simple indexing system for formulae, variables and Skolem functions. Moreover, the way in which variables are split enables us to restrict the term universe branchwise. Compared to standard sequent calculi or tableau systems with an eigenvariable condition our system is at least as good wrt. both proof length and the size of the search space, in addition to allowing full flexibility in the order of rule application.

A.1 INTRODUCTION

The aim of this paper is to present a solution to the problem of variable splitting in free variable systems for classical logic without equality. Attempting both to reveal the nature of this problem and to motivate our contribution, we shall start out by a brief discussion of the types of quantifier rules used in three standard proof systems for classical logic. To facilitate comparison we address their sequent calculi formulations. Following the notation of Smullyan [38] $\mathsf{L}\forall$ and $\mathsf{R}\exists$ are referred to as γ-rules, $\mathsf{L}\exists$ and $\mathsf{R}\forall$ are called δ-rules, branching rules are β-type rules and the remaining logical rules are of α-type. If θ is one of these types, an occurrence of θ-formula is a formula occurrence which potentially can be principal in a θ-type inference.

In comparing the complexity of search spaces one relevant factor is the length of proofs (counted in number of inference steps). Of no less importance, although much harder to measure, is the uniformity of the search space and the possibility of avoiding irrelevant steps in the search process.

The following four questions related to proof-search are among others central for this issue.

1. Is it possible to apply the rules in any order?
2. Does the system admit free variables and variable binding by means of explicit substitutions?
3. Given that the system has free variables and does not constrain the intrinsic order of rules, is the number of explicit copies of γ-formulae on a branch independent of the intrinsic rule order?
4. Can the number of explicit copies of γ-formulae on a given branch be locally bound by the term universe of the branch?

A negative answer to the first question implies that there are limited possibilities for goal-directed search, i.e. search driven by connections or potential axioms. It is then hard to prevent expansion of irrelevant formulae. If the answer to the second question is negative, we must choose instantiation terms along with applications of γ-type rules. We then run the risk of instantiating quantifiers with irrelevant terms, and this may give rise to irrelevant inferences. A positive answer to the first two questions greatly improves the possibility of performing *least commitment* search. Nevertheless, it may still be the case that unification-based goal-directed search has a cost in terms of proof length (addressed by the third and fourth questions). Should this be the case, the least commitment strategy may give rise to redundancies in the proof objects themselves, and the benefits of the strategy becomes harder to measure. The fourth question has partly to do with termination within decidable fragments of the language. An affirmative answer greatly facilitates the formulation of an efficient termination criterion. On the contrary, a negative answer is likely to have a negative impact both on the complexity of the search space and on the length of proofs (due to redundant inferences). We may also fail to detect that a given sequent is unprovable in cases where the term universe of one open branch would have been finitely bounded, given another set of quantifier rules.

Let us first address the standard Fregean treatment of quantifiers in terms of eigenparameters[1]. For systems without free variables, and which adopt Gentzen's eigenparameter condition, the answers to the first two questions are negative. However, the answer to the fourth is positive; in fact proofs in this family of Gentzen systems are in most cases very short[2].

If we let the γ-rules introduce free variables and the δ-rules introduce Skolem functions (the type of which is irrelevant here), we can delay the choice of instantiation terms to the level of axioms and select them on the basis of appropriate equations. In consequence, the rule dependencies expressed by the eigenparameter condition are replaced by term dependencies defined by unification problems. This gives systems with a positive answer to questions 1 and 2.

However, to say that γ-rules generate free variables is not a sufficient description of the free variable system. We must also specify a mechanism for selecting free variables, and this has an impact on the other two questions. In one extreme, we may select a fresh variable for each γ-rule application. This strategy generates *variable-pure* skeletons and is illustrated by the leftmost skeleton π_1 below. If, in that skeleton, the two variables u_1 and u_2 are distinct, the skeleton is variable-pure. Note that the skeleton can be extended to a proof without any new application of a $L\forall$ inference: the final skeleton can be closed by the substitution $\{u_1/a, u_2/b\}$. Also note that in each case the inference which introduces a variable occurs above the inference which introduces the Skolem function of the binding; this property corresponds to the fulfilment of the eigenparameter condition. In π_2 the order is reversed. Note that the free variable u is copied into the two branches, creating a dependency which is absent in π_1. In π_2 we must apply a $L\forall$ once more in one of the branches to close the skeleton.

[1] We shall use 'eigenparameter' instead of the more usual 'eigenvariable', since eigenterms are generated by δ-rules. The term 'parameter' stands for a constant or a 0-arity Skolem function and shall be used in relation to δ-rules, while the term 'free variable' is reserved for terms generated by γ-rules.

[2] Exceptions to this are addressed in section A.4.

$$\dfrac{\dfrac{\dfrac{\dfrac{u_1 = a}{\vdots}}{\forall x\varphi x, \varphi u_1 \vdash \xi a}}{\forall x\varphi x \vdash \xi a}\gamma_{u_1}}{\forall x\varphi x \vdash \forall x\xi x}\delta_a \quad \dfrac{\dfrac{\dfrac{u_2 = b}{\vdots}}{\forall x\varphi x, \varphi u_2 \vdash Qb}}{\dfrac{\forall x\varphi x \vdash Qb}{\forall x\varphi x \vdash \forall xQx}\delta_b}\gamma_{u_2}$$

$$\dfrac{\forall x\varphi x \vdash \forall x\xi x \qquad \forall x\varphi x \vdash \forall xQx}{\forall x\varphi x \vdash \forall x\xi x \wedge \forall xQx}\beta$$

Skeleton π_1

$$\dfrac{\dfrac{\dfrac{\begin{matrix}u = a\\ \vdots\\ \forall x\varphi x, \varphi u \vdash \xi a\end{matrix}}{\forall x\varphi x, \varphi u \vdash \forall x\xi x}\delta_a \quad \dfrac{\begin{matrix}u = b\\ \vdots\\ \forall x\varphi x, \varphi u \vdash Qb\end{matrix}}{\forall x\varphi x, \varphi u \vdash \forall xQx}\delta_b}{\forall x\varphi x, \varphi u \vdash \forall x\xi x \wedge \forall xQx}\beta}{\forall x\varphi x \vdash \forall x\xi x \wedge \forall xQx}\gamma_u$$

Skeleton π_2

In π_1 and π_2, φx is $\exists y(Pxy \wedge Qx)$ and ξx is $\exists yPxy$.

Variable-pure skeletons correspond to free variable tableaux [44]. As the example illustrates, the answer to question 3 is negative for these systems. The answer to question 4 is in general also negative, unless a rule ordering is chosen which is guaranteed to fulfil the eigenparameter condition. However, question 1 then receives a negative answer. To remedy this situation a strategy for identifying universal variables was proposed in [9]. Applied to π_2, the strategy identifies the occurrences of u as universal in the branches; u can then be bound to different terms in different branches. However, even if this idea works in this particular example, it has limited range. It does e.g. not work for the sequent $\forall x(Px \to (Qx \wedge Rx)) \vdash \forall x(Px \to Qx) \wedge \forall x(Px \to Rx)$ given eager use of the $L\forall$ rule.

If the variables u_1 and u_2 in π_1 are identical, the skeleton is *variable-sharing*. This class of skeletons was identified in [44], where it is shown that leaf sequents of skeletons which are balanced (defined below) correspond to paths through matrices [12]. As this strategy for selecting variables generates freely permuting skeletons, the answer to question 3 is positive. However, since the price of the nice permutation properties is strong dependencies among variable occurrences, question 4 receives a negative answer.

Attempting to solve the redundancy problem Bibel sketched an idea for variable splitting [11]. We believe that the system introduced in this paper can be taken as a sharp formulation of his idea, fully generalized to non-clausal formulae. This is achieved by dynamically renaming variables uniformly over the skeleton as the skeleton is developed. Technically, the formulation exploits a fairly simple system of indices. As skeletons of our system are freely permuting, the answers to the first three questions is the same as for variable-sharing (i.e. matrix) systems. And since we can fully simulate proofs constructed in a calculus with eigenparameters, question 4 receives the same answer as for this calculus. We hence combine the best of the three quantifier treatments discussed in this section and can respond 'yes' to all four questions.

The focus of the current paper is exclusively on the class of balanced skeletons, which is the class corresponding to matrices. This restriction can however be avoided at the cost of a more complex syntax.

A.2 The free variable system

The object language contains the usual quantifiers and connectives, but it also contains a stock of indices of various sorts. It is essential that these constructs are part of the language itself; they are not just meta-language devices used to talk about formulae.

The set of closed first-order *formulae* is inductively defined from a fixed, countable set of predicate symbols and a set of *quantification variables*, which only occur bound by quantifiers. The logical operators are $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\forall$ and $\exists$. In addition, we allow formulae to contain instantiation terms. *Instantiation terms* are inductively defined from countable sets $\mathcal{U} = \{u_m^n \mid m, n \in \mathbb{N}\}$ of *instantiation variables* and $\mathcal{F} = \{f_m^n \mid m, n \in \mathbb{N}\}$ of *Skolem function symbols*. We require that instantiation terms do not contain quantification variables. The purpose of instantiation terms is to provide a syntax for free variables and run-time skolemization. Formulae with instantiation terms will be generated by the rules of the calculus and do not exist outside such a context. Furthermore, we shall distinguish 'formulae' from 'formula occurrences'. A *formula occurrence* is a formula in which each subformula, including the formula itself, is augmented by an *index pair* ${}^n_m$; the subscript m is called an *occurrence number* and the superscript n is called *copy number*. Each formula occurrence, but not its subformulae, is also labelled with a *splitting set* S, which is a set of index pairs.

Example A.1 Here is an example of a formula occurrence:

$$(\exists y((Pu_1^2 y)_4^2 \wedge (Qu_1^2)_5^2)_3^2)_2^2\{{}^1_7, {}^1_8\}$$

The purpose of the occurrence numbers is to uniquely identify all subformulae occurrences in a given structure. Therefore, in a formula occurrence, all subformula occurrences must have distinct occurrence numbers. An occurrence number can be thought of as a pointer to a position in a formula tree [46] or to a memory location.

The copy numbers correspond to Bibel's notion of *multiplicity* [11]; we are thus able to speak of two distinct occurrences of the same formula. For a given formula occurrence, the copy number must be the same for all subformula occurrences. This number will be incremented as new explicit copies of a formula are made by the implicit contraction in γ-rules.

The purpose of the splitting set is to keep track of the β-formulae that have split the skeleton into branches, in order to split the instantiation variables accordingly.

The system manipulates *sequents* of the form $\Gamma \vdash \Delta$, where Γ and Δ are sets of formula occurrences. A *root sequent* is a sequent in which:

- all occurrence numbers are distinct
- all copy numbers are identical to 1
- all terms are quantification variables, bound by quantifiers
- all splitting sets are empty

Example A.2 Here is the root sequent discussed in section 1:

$$(\forall x(\exists y((Pxy)^1_4 \wedge (Qx)^1_5)^1_3)^1_2)^1_1\{\} \vdash ((\forall x(\exists y(Pxy)^1_9)^1_8)^1_7 \wedge (\forall x(Qx)^1_{11})^1_{10})^1_6\{\}$$

Since the system of index pairs makes a formula hard to read, we will allow for abbreviations. We can denote the root sequent like this:

$$\underset{1}{\forall x}\underset{2}{\exists y}(\underset{4}{Pxy} \underset{3}{\wedge} \underset{5}{Qx})^1_1 \vdash (\underset{7}{\forall x}\underset{8}{\exists y}\underset{9}{Pxy} \underset{6}{\wedge} \underset{10}{\forall x}\underset{11}{Qx})^1_6$$

A *skeleton* is a finitely branching tree generated from a root sequent and the rules given in Fig. A.1. The formulae in Γ and Δ are called *extra formulae*, while the other formula in the conclusion is called the *principal formula* and the other formula(e) in the premiss(es) are called *active* formulae.

Meta-language conventions: In $\varphi^n_m S$, m denotes the occurrence number given to φ, n is the copy number given to all subformula occurrences of φ and S is the splitting set.

Explanations:

β-rules: $\Gamma \uplus^l_k$ denotes $\{\varphi^n_m S \cup \{^l_k\} \mid \varphi^n_m S \in \Gamma\}$, the set of formula occurrences in Γ where the index pair ^l_k has been added to the splitting sets. The β-rules split the skeleton into branches and add the corresponding index pairs to the splitting sets of all other formula occurrences than the β-formula. If the β-formula in question is $(\varphi^n_k \wedge \psi^n_l)^n_m S$, then the index pairs ^n_k and ^n_l are called *dual*. The dual index pair of ^n_m will be denoted $\frac{n}{m}$.

γ-rules: These rules introduce instantiation variables u^n_m, where ^n_m corresponds to the index pair of the principal formula occurrence. In addition, α-rules

$$\frac{\Gamma, \varphi_k^n S, \psi_l^n S \vdash \Delta}{\Gamma, (\varphi_k^n \wedge \psi_l^n)_m^n S \vdash \Delta} \ \mathsf{L}\wedge$$

$$\frac{\Gamma \vdash \varphi_k^n S, \psi_l^n S, \Delta}{\Gamma \vdash (\varphi_k^n \vee \psi_l^n)_m^n S, \Delta} \ \mathsf{R}\vee$$

$$\frac{\Gamma, \varphi_k^n S \vdash \psi_l^n S, \Delta}{\Gamma \vdash (\varphi_k^n \rightarrow \psi_l^n)_m^n S, \Delta} \ \mathsf{R}\!\rightarrow$$

$$\frac{\Gamma, \varphi_k^n S \vdash \Delta}{\Gamma \vdash (\neg\varphi_k^n)_m^n S, \Delta} \ \mathsf{R}\neg$$

$$\frac{\Gamma \vdash \varphi_k^n S, \Delta}{\Gamma, (\neg\varphi_k^n)_m^n S \vdash \Delta} \ \mathsf{L}\neg$$

β-rules

$$\frac{\Gamma \uplus {}^n_k \vdash \varphi_k^n S, \Delta \uplus {}^n_k \quad \Gamma \uplus {}^n_l \vdash \psi_l^n S, \Delta \uplus {}^n_l}{\Gamma \vdash (\varphi_k^n \wedge \psi_l^n)_m^n S, \Delta} \ \mathsf{R}\wedge$$

$$\frac{\Gamma \uplus {}^n_k, \varphi_k^n S \vdash \Delta \uplus {}^n_k \quad \Gamma \uplus {}^n_l, \psi_l^n S \vdash \Delta \uplus {}^n_l}{\Gamma, (\varphi_k^n \vee \psi_l^n)_m^n S \vdash \Delta} \ \mathsf{L}\vee$$

$$\frac{\Gamma \uplus {}^n_k \vdash \varphi_k^n S, \Delta \uplus {}^n_k \quad \Gamma \uplus {}^n_l, \psi_l^n S, \vdash \Delta \uplus {}^n_l}{\Gamma, (\varphi_k^n \rightarrow \psi_l^n)_m^n S \vdash \Delta} \ \mathsf{L}\!\rightarrow$$

δ-rules

$$\frac{\Gamma \vdash \varphi[x/f_m^n(\vec{u})]_k^n S, \Delta}{\Gamma \vdash (\forall x \varphi_k^n)_m^n S, \Delta} \ \mathsf{R}\forall$$

$$\frac{\Gamma, \varphi[x/f_m^n(\vec{u})]_k^n S \vdash \Delta}{\Gamma, (\exists x \varphi_k^n)_m^n S \vdash \Delta} \ \mathsf{L}\exists$$

γ-rules

$$\frac{\Gamma, (\forall x \varphi_k^{n+1})_m^{n+1} S, \varphi[x/u_m^n]_k^n S \vdash \Delta}{\Gamma, (\forall x \varphi_k^n)_m^n S \vdash \Delta} \ \mathsf{L}\forall$$

$$\frac{\Gamma \vdash (\exists x \varphi_k^{n+1})_m^{n+1} S, \varphi[x/u_m^n]_k^n S, \Delta}{\Gamma \vdash (\exists x \varphi_k^n)_m^n S, \Delta} \ \mathsf{R}\exists$$

Figure A.1: The rules of the free variable system

the γ-rules have built-in an implicit contraction operation and introduce new contracted copies of the principal formula occurrence. The new occurrence is obtained from the principal occurrence simply by incrementing the copy number.

δ-rules: These rules skolemize the principal formula occurrences in the following way: If $(\forall x \varphi_k^n)_m^n S$ is the principal formula occurrence in which exactly the instantiation variables $\vec{u} = u_{m_1}^{n_1}, \ldots, u_{m_i}^{n_i}$ occur, then the Skolem term $f_m^n(\vec{u})$ is introduced and replaced for the variable x. This δ-rule lies somewhere between a δ^+-rule [28] and a δ^{+^+}-rule [10]. It is δ^+-like in the sense that only variables in the current formula occurrence matter, not all variables in the whole branch, like the original δ-rule, or all relevant variables, like the δ^*-rule [16]. Moreover, all formula occurrences with the same index pair will introduce identical Skolem function symbols, which is more δ^{++}-like, at least with respect to different branches. (A closer approximation to the δ^{++}-rule could be obtained by skipping the copy numbers of the Skolem function symbols altogether.)

A skeleton is composed out of instances of rules, called *inferences*. An inference can be uniquely identified by its principal formula occurrence. We can therefore denote the inferences in a skeleton with the inference type and the index pair of the principal formula, like this: θ_m^n, where θ is the inference type, m is the occurrence number and n is the copy number of the principal formula occurrence. We use r and s as metasymbols for inferences.

Skeletons can be represented at a higher level of abstraction by diagrams in which the diagram labels denote inferences and arrows denote relations between inferences. The following are diagram labels:

α-inference:	$\circ$	γ-inference:	γ_m^n
β-inference:	$\triangle$	δ-inference:	δ_m^n

First, we need to define the immediate ancestor relation. The inference r is an *immediate ancestor* of the inference s, written $r \gg s$, if the principal formula occurrence of r has the same occurrence and copy number as an active formula occurrence of s. The splitting sets of the two occurrences may be different; the splitting set of r might be a superset of the splitting set of s. It is necessary that the inference r is above s in the skeleton. (In the diagrams, two $\gg$-related inferences will have a similar looking arrow between them.)

If r is an inference in a skeleton, then the $\gg$*-tree* rooted in r is the least tree T such that $r \in T$ and for all $s \in T$, if $s' \gg s$, then $s' \in T$.

Two different inferences r and s are *contextually equivalent*, written $r \sim s$, if the index pairs of r and s are identical. In the diagrams, two contextually equivalent relations will have dots between them.

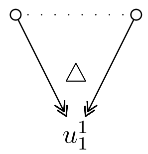
A *balanced* skeleton is one in which the following condition holds for all inferences r and s in the skeleton: If $r \sim s$ and $r' \gg r$, then there is an s' such that $s' \gg s$ and $r' \sim s'$.

We will from now on require all skeletons to be balanced. This is the class of skeletons which corresponds directly to matrices[44, Section 2.5].

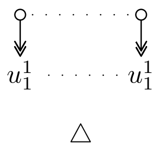
A *permutation variant* of a skeleton is a skeleton which differ only in the *order* of rule applications.

The following two skeletons are permutation variants. In the uppermost skeleton, the γ-inference has been applied before the β-inference; in the lowermost skeleton it is the other way around. Notice that the situation is symmetrical; the two inferences directly above the β-inference in the lowermost skeleton are contextually equivalent. Only indices and splitting sets of particular relevance for the example are displayed. Empty splitting sets and implicit copies of γ-formulae are omitted.

$$\dfrac{\dfrac{\dfrac{(Pu^1_1)\{{}^1_2\},(Qu^1_1)\{{}^1_2\}\vdash\varphi^1_2}{(Pu^1_1\wedge Qu^1_1)\{{}^1_2\}\vdash\varphi^1_2}\qquad\dfrac{(Pu^1_1)\{{}^1_3\},(Qu^1_1)\{{}^1_3\}\vdash\psi^1_3}{(Pu^1_1\wedge Qu^1_1)\{{}^1_3\}\vdash\psi^1_3}}{Pu^1_1\wedge Qu^1_1\vdash\varphi^1_2\wedge\psi^1_3}\,\beta}{\forall x(Px\wedge Qx)^1_1\vdash\varphi^1_2\wedge\psi^1_3}\,\gamma^1_1$$



$$\dfrac{\dfrac{\dfrac{(Pu^1_1)\{{}^1_2\},(Qu^1_1)\{{}^1_2\}\vdash\varphi^1_2}{(Pu^1_1\wedge Qu^1_1)\{{}^1_2\}\vdash\varphi^1_2}}{\forall x(Px\wedge Qx)^1_1\{{}^1_2\}\vdash\varphi^1_2}\,\gamma^1_1\qquad\dfrac{\dfrac{(Pu^1_1)\{{}^1_3\},(Qu^1_1)\{{}^1_3\}\vdash\psi^1_3}{(Pu^1_1\wedge Qu^1_1)\{{}^1_3\}\vdash\psi^1_3}}{\forall x(Px\wedge Qx)^1_1\{{}^1_3\}\vdash\psi^1_3}\,\gamma^1_1}{\forall x(Px\wedge Qx)^1_1\vdash\varphi^1_2\wedge\psi^1_3}\,\beta$$



Examples like this are easily generalized to general permutation schemes, according to which the γ-inferences are symmetrically interchanged with the β-inferences. By inspecting the patterns of these schemes, it is straightforward to verify the following key property for balanced skeletons:

A.3 Lemma (Invariance lemma) The sets of leaf sequents are invariant under permutation. Any two permutation variants have identical leaf sequents. $\dashv$

A.4 Lemma (Permutation lemma) For any sub-skeleton with $\Gamma, \varphi^n_m S \vdash \Delta$ in the root, $\varphi^n_m S$ non-atomic with an ancestor of $\varphi^n_m S$ expanded somewhere in the sub-skeleton, there is a permutation variant of the sub-skeleton which has $\varphi^n_m S$ as the principal formula occurrence in the lowermost inference. $\dashv$

Proof. Let $S \subseteq S'$. Since $\varphi^n_m S'$ is principal somewhere in the sub-skeleton, there is a set of contextually equivalent inferences in the skeleton for which the principal formulae occurrences have the form $\varphi^n_m T$ for $S \subseteq T$. All of these inferences must occur in different branches. By repeatedly choosing the inferences which are furthest away from the root and permuting according to permutation schemes, we obtain a skeleton in which $\varphi^n_m S$ is the principal formula. The schemas apply since the skeleton is balanced [44, Lemma 2.14]. $\square$

A.3 Connections, colorings and proofs

The purpose and intuition behind the splitting sets is to identify instantiation variables by the splitting sets of the surrounding formula occurrences. The uniformity of the splitting sets provides a machinery to split variables maximally without losing sight of logical dependencies between these. However, the splitting mechanism in the rules is in some cases too liberal. To compensate for this the logical dependencies of some variables will be regained by introducing term equations which identify certain instantiation variables. We wish to define proofs as skeletons in which all leaf nodes in all branches are closed by a unifying substitution. To do this, we must first define precisely how the splitting of terms is achieved and then specify the conditions which the substitutions must satisfy. This can be defined altogether at the level of terms, since all permutation variants of the underlying skeleton generate the same set of leaves.

A *color* is a splitting set occurring in a leaf sequent.

If $P(s_1, \ldots, s_k)^n_m S$ is an atomic formula occurrence and s_i is one of the instantiation terms, then the color S gives the corresponding *colored term* $s_i \oplus S$ like this:

- $u^n_m \oplus S = u^n_m S$
- $f^n_m(t_1, \ldots, t_l) \oplus S = f^n_m S(t_1 \oplus S, \ldots, t_l \oplus S)$

sT stands for the colored term $s \oplus T$

$P(s_1 \oplus S, \ldots, s_k \oplus S)$ is the corresponding *colored formula*.

If $P(s_1 \ldots s_k)^{n_1}_{m_1} S_1 \vdash P(t_1 \ldots t_n)^{n_2}_{m_2} S_2$ is a subsequent of a leaf sequent in a skeleton π, then the pair of colored formulae $P(s_1 \oplus S_1, \ldots, s_k \oplus S_1) \vdash P(t_1 \oplus S_2, \ldots, t_n \oplus S_2)$ is called a *connection*. Each connection c has an associated set $\Pi(c)$ of equations, where $\Pi(c)$ contains exactly the equations of the form $u^n_m S = u^n_m T$ such that $u^n_m S$ occurs in the connection and $u^n_m T$ occurs in the same sequent which gives rise to the connection. A set of connections C, typically one connection for each branch in the skeleton, defines the set of *colored variables* of C, containing all colored instantiation variables occurring either in a connection c in C or in an associated set of equations $\Pi(c)$, and a set of *colored function symbols*, containing all colored function symbols occurring in a connection in C. The *Herbrand universe* of C contains all colored terms that can be generated from these.

Let C be a set of connections. A *substitution* is a function σ which maps colored variables of C into the Herbrand universe of C. It can be partial; the domain is $\mathsf{dom}(\sigma)$. Important properties of a substitution σ are:

- σ is *idempotent* if for all colored variables t of C, $\sigma(t) = \sigma(\sigma(t))$
- σ is *closing* if for each connection $c \in C$, $c\sigma$ is of the form $\varphi \vdash \varphi$, and for all equations $v = w \in \Pi(c)$, $\sigma(v) = \sigma(w)$.
- σ is *homogeneous* if the following condition holds: If $\sigma(u^l_k S) = tT$ and $^n_m \in T \cap S$, then $u^l_k S[^n_m/^n_{\overline{m}}]$ is in $\mathsf{dom}(\sigma)$ and $\sigma(u^l_k S[^n_m/^n_{\overline{m}}]) = tT[^n_m/^n_{\overline{m}}]$. ($S[^n_m/^n_{\overline{m}}]$ is the splitting set S where ^n_m has been replaced by its *dual* index pair $^n_{\overline{m}}$.)

The intuition behind the homogeneity property is the following: The fact that $^n_m \in T \cap S$ means that the β-inference whose one active formula is marked ^n_m does *not* contribute to the closing of the branch in question. Then, it should not, symmetrically, contribute to the closure of the corresponding leaf at the other side of the β-inferences either. (See Ex. A.7 for an example of what can happen without this condition.)

Let π be a skeleton. A set C of connection is *spanning* for π if there is exactly one connection in C for each branch in π.

Let C be spanning for a skeleton π. Then, $\langle \pi, C, \sigma \rangle$ is a *proof* of the root sequent in π if σ is idempotent, homogeneous and closing.

A.5 Lemma If $\langle \pi, C, \sigma \rangle$ is a proof and π' is a permutation variant of π, then $\langle \pi', C, \sigma \rangle$ is also a proof. Hence, colors are invariant under permutation. $\dashv$

Proof. By Lemma A.3, the leaf nodes of π' are identical to the leaf nodes of π. Thus, the colors are invariant under permutation. Furthermore, since C and σ do not change, σ is still idempotent, homogeneous and closing. $\square$

We now lift the notion of colors of terms and formulae to the level of *colored inferences*. If $t^n_m S$ is a colored term in a connection, then there will be a unique θ-inference θ^n_m in the skeleton which introduces the instantiation term t^n_m. Of course, there can be many θ^n_m-inferences in the skeleton, but only one will be below the leaf sequent which gives rise to the given connection. $\theta^n_m S$ denotes the θ^n_m-inference colored with S. One θ-inference can be colored in many different ways, due to different splitting sets in different connections. Note that only γ- and δ-inferences are colored.

A spanning set C of connections gives rise to a *colored skeleton*, a skeleton in which some, but not necessarily all, quantifier inferences are colored according to the colored terms in C. Notice that if an inference r is colored rS and $r \sim r'$, then r' is *not* colored $r'S$.

Let $\langle \pi, C, \sigma \rangle$ be a proof and assume that σ is of minimal cardinality. The substitution σ induces a relation $\dashrightarrow_\sigma$ defined in the following way: If

$\sigma(u^n_m S) = tT$ and rS and $r'T$ are the corresponding colored inferences, both occurring on the same branch of the skeleton, then $rS \dashrightarrow_\sigma r'T$.

In diagrams for colored skeletons we can now draw $\dashrightarrow_\sigma$-arrows between the inferences to visualize the relations induced from a closing substitution. The arrows will be labelled with splitting sets to display which colored inference that are $\dashrightarrow_\sigma$-related.

Example A.6 Let π be a skeleton generated from the root sequent given in Ex. A.2 above. Assume that the lowermost inference is $\mathsf{R}\wedge$:

$$\dfrac{\forall x\exists y(Pxy \wedge Qx)^1_1\{{}^1_7\} \vdash (\forall x\exists yPxy)^1_7 \quad \forall x\exists y(Pxy \wedge Qx)^1_1\{{}^1_{10}\} \vdash (\forall xQx)^1_{10}}{\forall x\exists y(Pxy \wedge Qx)^1_1 \vdash (\forall x\exists yPxy \wedge \forall xQx)^1_6}\ \mathsf{R}\wedge$$

The left branch gives rise to the connection $Pu^1_1\{{}^1_7\}f^1_2\{{}^1_7\}(u^1_1\{{}^1_7\}) \vdash Pf^1_7u^1_8$.

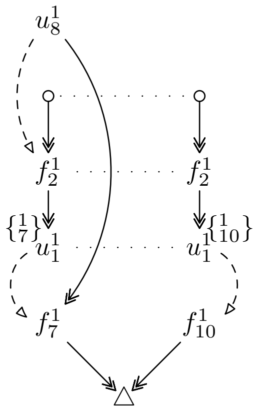
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(Pu^1_1f^1_2u^1_1)^1_4\{{}^1_7\}, (Qu^1_1)^1_5\{{}^1_7\} \vdash (Pf^1_7u^1_8)^1_9}{(Pu^1_1f^1_2u^1_1)^1_4\{{}^1_7\}, (Qu^1_1)^1_5\{{}^1_7\} \vdash (\exists yPf^1_7y)^1_8}\ \gamma^1_8}{(Pu^1_1f^1_2u^1_1 \wedge Qu^1_1)^1_3\{{}^1_7\} \vdash (\exists yPf^1_7y)^1_8}}{\exists y(Pu^1_1y \wedge Qu^1_1)^1_2\{{}^1_7\} \vdash (\exists yPf^1_7y)^1_8}\ \delta^1_2}{\forall x\exists y(Pxy \wedge Qx)^1_1\{{}^1_7\} \vdash (\exists yPf^1_7y)^1_8}\ \gamma^1_1}{\forall x\exists y(Pxy \wedge Qx)^1_1\{{}^1_7\} \vdash (\forall x\exists yPxy)^1_7}\ \delta^1_7$$

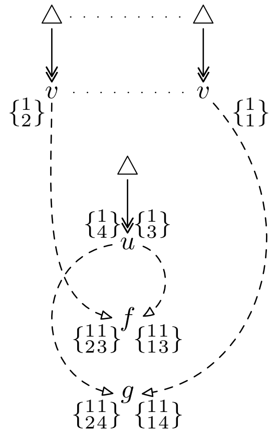
The right branch gives rise to the connection $Qu^1_1\{{}^1_{10}\} \vdash Qf^1_{10}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{(Pu^1_1f^1_2u^1_1)^1_4\{{}^1_{10}\}, (Qu^1_1)^1_5\{{}^1_{10}\} \vdash (Qf^1_{10})^1_{11}}{(Pu^1_1f^1_2u^1_1 \wedge Qu^1_1)^1_3\{{}^1_{10}\} \vdash (Qf^1_{10})^1_{11}}}{\exists y(Pu^1_1y \wedge Qu^1_1)^1_2\{{}^1_{10}\} \vdash (Qf^1_{10})^1_{11}}\ \delta^1_2}{\forall x\exists y(Pxy \wedge Qx)^1_1\{{}^1_{10}\} \vdash (Qf^1_{10})^1_{11}}\ \gamma^1_1}{\forall x\exists y(Pxy \wedge Qx)^1_1\{{}^1_{10}\} \vdash (\forall xQx)^1_{10}}\ \delta^1_{10}$$

The substitution $\sigma = \{u^1_1\{{}^1_7\} \mapsto f^1_7, u^1_8 \mapsto f^1_2\{{}^1_7\}(f^1_7), u^1_1\{{}^1_{10}\} \mapsto f^1_{10}\}$ is closing for the two connections and provides a proof.

Below to the left is the skeleton diagram. To the right is the skeleton diagram for Example A.7, depicting a substitution which is *not* a proof. The skeletons diagrams have arrows displaying the immediate ancestor relation and the $\dashrightarrow_\sigma$-relation.





Example A.7 $(\forall x((Px)^1_1 \lor (Qx)^1_2))^1_5, (\forall x((Px)^1_3 \lor (Qx)^1_4))^1_6 \vdash (\forall x Px)^1_7, (\forall x Qx)^1_8$ is not a valid sequent, but without the requirement that substitutions are homogeneous, it would be provable. Below to the left are the generated leaf sequents (v,w,f,g abbreviate u^1_5, u^1_6, f^1_7, f^1_8, respectively). Below to the right is a closing, non-homogeneous, substitution σ for a spanning set of connections.

$$\begin{array}{ll}
Pv\{\genfrac{}{}{0pt}{}{1}{3}\}, Pw\{\genfrac{}{}{0pt}{}{1}{1}\} \vdash Pf\{\genfrac{}{}{0pt}{}{11}{13}\}, Qg\{\genfrac{}{}{0pt}{}{11}{13}\} & \quad v\{\genfrac{}{}{0pt}{}{1}{3}\} \mapsto f\{\genfrac{}{}{0pt}{}{11}{13}\} \\
Pv\{\genfrac{}{}{0pt}{}{1}{4}\}, Qw\{\genfrac{}{}{0pt}{}{1}{1}\} \vdash Pf\{\genfrac{}{}{0pt}{}{11}{14}\}, Qg\{\genfrac{}{}{0pt}{}{11}{14}\} & \quad w\{\genfrac{}{}{0pt}{}{1}{1}\} \mapsto g\{\genfrac{}{}{0pt}{}{11}{14}\} \\
Qv\{\genfrac{}{}{0pt}{}{1}{3}\}, Pw\{\genfrac{}{}{0pt}{}{1}{2}\} \vdash Pf\{\genfrac{}{}{0pt}{}{11}{23}\}, Qg\{\genfrac{}{}{0pt}{}{11}{23}\} & \quad w\{\genfrac{}{}{0pt}{}{1}{2}\} \mapsto f\{\genfrac{}{}{0pt}{}{11}{23}\} \\
Qv\{\genfrac{}{}{0pt}{}{1}{4}\}, Qw\{\genfrac{}{}{0pt}{}{1}{2}\} \vdash Pf\{\genfrac{}{}{0pt}{}{11}{24}\}, Qg\{\genfrac{}{}{0pt}{}{11}{24}\} & \quad v\{\genfrac{}{}{0pt}{}{1}{4}\} \mapsto g\{\genfrac{}{}{0pt}{}{11}{24}\}
\end{array}$$

It is not homogeneous, since $u\{\genfrac{}{}{0pt}{}{1}{3}\} \mapsto f\{\genfrac{}{}{0pt}{}{11}{13}\}$, but it is *not* the case that $u\{\genfrac{}{}{0pt}{}{1}{4}\} \mapsto f\{\genfrac{}{}{0pt}{}{11}{14}\}$. The index pair $\frac{1}{3}$ occurs in both $u\{\genfrac{}{}{0pt}{}{1}{3}\}$ and $f\{\genfrac{}{}{0pt}{}{11}{13}\}$, and since $\frac{1}{3}$ and $\frac{1}{4}$ are dual, it is required that $u\{\genfrac{}{}{0pt}{}{1}{4}\}$ is in the domain of σ, which it is, *and* that it is mapped to $f\{\genfrac{}{}{0pt}{}{11}{14}\}$, which it is not, since it is mapped to $g\{\genfrac{}{}{0pt}{}{11}{24}\}$.

A.4 Consistency

The consistency argument contains two parts of a very different nature. The first part demonstrates that if a certain skeleton relation (the reduction ordering) is acyclic, the skeleton can be permuted into an object with the σ-conformity property (defined below). Proofs with this property naturally correspond to proof objects in a calculus with quantifier rules which observe the eigenparameter condition. Incidentally, the fact that this also holds the other way around shows the completeness of the free variable system. The second part of the consistency argument shows that any proof can be extended to a proof for which the associated reduction ordering is acyclic. The extension can in odd cases give an exponential increase in proof length.

Let us first make some of the notions precise. A colored inference rS is *projected* onto a unique skeleton inference r. Note that two colored inferences can be projected onto the same skeleton inference. The *reduction ordering* $\rhd$ on the skeleton is induced by the substitution in the following way: $r \rhd s$ iff either $r \gg s$ or there are colored inferences rS and sT projected onto r and s such that $rS \dashrightarrow_\sigma sT$.

Let $\langle \pi, C, \sigma \rangle$ be a proof. π *conforms to* σ if the following two conditions hold:

(1) for all colored γ-inferences rS and $r'S'$ with identical projections, if $rS \dashrightarrow_\sigma sT$ and $r'S' \dashrightarrow_\sigma s'T'$, then sT and $s'T'$ have identical projections,

(2) if $r \rhd s$, then r is above s in the skeleton.

Relating the condition (2) to rules in a calculus with explicit instantiations of eigenparameters in γ-inferences (cnf. Section. 1), it corresponds to the fulfillment of an eigenparameter condition for relevant instantiations. It also corresponds to the condition that a δ-inference must introduce a term which does not occur in the conclusion. The function of (1) is to secure that a particular γ-inference receives a unique instantiation on a given branch.

From the immedate ancestor relation (which is defined only for non-colored inferences), we define a similar relation for *colored* inferences in a *colored* skeleton π: If either $r \gg^+ r'$ or there is an inference r'' such that $r \gg^+ r''$ and $r'' \sim r'$, where $\gg^+$ is the transitive closure of $\gg$, then $rS \twoheadrightarrow r'T$. (The inference r'' does not need to be colored.)

A *cycle* is a sequence of colored inferences $r_1R_1, s_1S_1, \ldots, r_nR_n, s_nS_n$ such that

$$\begin{array}{c} r_1R_1 \dashrightarrow_\sigma s_1S_1 \twoheadrightarrow r_2R_2 \\ r_2R_2 \dashrightarrow_\sigma s_2S_2 \twoheadrightarrow r_3R_3 \\ \ldots \\ r_nR_n \dashrightarrow_\sigma s_nS_n \twoheadrightarrow r_1R_1 \end{array}$$

A proof is *cycle-free* there is no cycle through its colored inferences. It is immediate that a proof is cycle-free iff the associated reduction ordering is cycle-free.

A.8 Theorem (Eigenparameter property theorem)

If a proof $\langle \pi, C, \sigma \rangle$ is cycle free, there is a permutation variant π' of π such that π' conforms to σ. $\dashv$

The proof of this theorem makes use of general properties of colors, formulated below. To see the content of these statements, recall that colors are developed branchwise by tracing the generation of a leaf formula. Its index pairs can be identified by repeatedly "selecting one side" of β-inferences of the branch whose principal formulae do not descend from the leaf formula occurrence which carries the color. The following notions characterize situations where colors have emerged by symmetrically choosing "different sides" of β-inferences; the β-inferences up to contextual equivalence.

Let S and T be colors. The *complementary image* of T in S is the set of all index pairs $\frac{n}{m} \in S$ such that $\frac{n}{\overline{m}} \in T$. S and T are *complementary* if $S \setminus T$ is the complementary image of T in S and $T \setminus S$ is the complementary image of S in T.

A.9 Lemma (Existence of complementary colors) Let uS and uT be two colorings of the same instantiation variable u. Then there is a leaf which contains both uS and $u\overline{T}$ for a color $\overline{T}$ complementary to T such that the complementary image of T in $\overline{T}$ is identical to the complementary image of T in S. ⊣

PROOF. Follows by observing that all β inferences identified by the complementary image of T in S have contextual equivalents in a branch which contains uS. □

A.10 Lemma (Generalized homogeneity property) Let r be a γ-type inference and s be either γ-type or δ-type. Let $rS \dashrightarrow_\sigma sT$. Then there is a set $\overline{S}$ complementary to S and a set $\overline{T}$ complementary to T such that $r\overline{S} \dashrightarrow_\sigma s\overline{T}$. Furthermore, both the complementary image of $\overline{S}$ in S and the complementary image of $\overline{T}$ in T is identical to $S \cap T$. ⊣

PROOF. Repeated application of the homogeneity property. □

A substitution σ has the generalized homogeneity property when it is the case that $\sigma(rS) = sT$ implies that $r\overline{S} \in \mathsf{dom}(\sigma)$ and $\sigma(r\overline{S}) = t\overline{T}$, where $\overline{S} := S[\frac{n_1}{m_1}/\frac{n_1}{\overline{m}_1}, \ldots, \frac{n_i}{m_i}/\frac{n_i}{\overline{m}_i}]$ and $\overline{T} := T[\frac{n_1}{m_1}/\frac{n_1}{\overline{m}_1}, \ldots, \frac{n_i}{m_i}/\frac{n_i}{\overline{m}_i}]$, for all $\{\frac{n_1}{m_1}, \ldots, \frac{n_i}{m_i}\} \subseteq S \cap T$.

We now return to the proof of the Eigenparameter property theorem, which is by branchwise induction on the number of inferences from the root sequent and upwards. Let us say that all inferences initially are *black*, and use as induction hypothesis that all white inferences satisfy the σ-conforming property. Select an inference r in the branch which is $\rhd$-minimal among the black inferences (this is the point which requires that the proof is cycle-free). Apply the Permutation lemma to the sequent above the uppermost white inference (initially the root sequent) to transfer r to a place where all inferences below r are white. Note that no white inferences are touched by the permutation process, so that we can apply the induction hypothesis to these inferences also for the new skeleton. Now turn r white. The second property of σ-conformity follows immediately from the induction hypothesis for r. If r is a γ-type inference, we must also verify the first property. So assume that rS and $r'S'$ have identical projections and that $rS \dashrightarrow_\sigma sT$ and $r'S' \dashrightarrow_\sigma s'T'$. If $S \neq S'$, Lemma A.9 gives the existence of variables uS and $u\overline{S'}$ such that $uS = u\overline{S'}$ must be an equation in the associated set of equations. Moreover, $\overline{S'}$ is complementary to S' and satisfies the conditions which enable us to apply Lemma A.10 and conclude that sT and $s'T'$ have identical projections. For sT and $s'T'$ project onto inferences which are either identical or contextually equivalent, but since they both occur below r, they must be identical.

A.11 Theorem (Cycle elimination theorem)

A proof $\langle \pi, C, \sigma \rangle$ can be extended to a proof which is cycle-free. $\dashv$

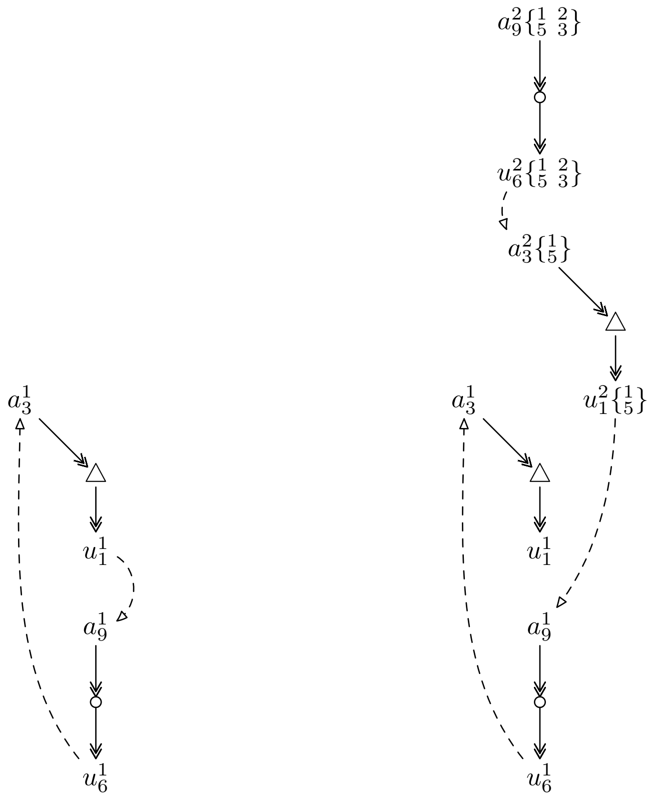
The proof of this result, which yields the consistency of the system, goes by branchwise induction on the number of cycles with colored inferences projected onto the branch. We provide a sketch of the argument. First, using homogeneity, one can show that a cycle

$$\begin{gathered} r_1R_1 \dashrightarrow_\sigma s_1S_1 \twoheadrightarrow r_2R_2 \\ \cdots \\ r_nR_n \dashrightarrow_\sigma s_nS_n \twoheadrightarrow r_1R_1 \end{gathered}$$

witness the existence of a *local* cycle, i.e. a cycle of the above form with the property that $R_1 \cap S_1 = R_1 \cap R_2, \ldots, R_n \cap S_n = R_n \cap R_1$. We next show a property about the γ-formulae φ_i, instances of which occur as principal in the colored γ-inferences r_iR_i. If we select a branch with a connection which gives rise to one $\dashrightarrow_\sigma$-arrow in the cycle, the leaf of this branch must contain an instance of each φ_i with a higher copy number. Using these instances and the locality property we can effectively add inferences to the branch so that all connections which give rise to a $\dashrightarrow_\sigma$-arrow in the cycle can be recreated (differing from the original only in copy numbers). The newly created cycle can, however, be broken by keeping one connection formula generated from inferences of the old cycle and taking the other connection formula from the newly created cycle. The process is illustrated in the following example.

Example A.12 Let the root sequent be: $\underset{1}{\forall x}(\underset{3}{\forall x}\underset{4}{Qx} \underset{2}{\rightarrow} \underset{5}{Px}) \vdash \underset{6}{\exists x}(\underset{8}{Qx} \underset{7}{\rightarrow} \underset{9}{\forall x}\underset{10}{Px})$

To the right is a skeleton diagram which is generated from the above root sequent and which contains the cycle $\gamma_6^1 \dashrightarrow_\sigma \delta_3^1 \twoheadrightarrow \gamma_1^1 \dashrightarrow_\sigma \delta_9^1 \twoheadrightarrow \gamma_6^1$. Eliminating the cycle yields the skeleton represented by the rightmost skeleton diagram:



The arrow $\gamma_1^1 \dashrightarrow_\sigma \delta_9^1$ is resolved by expanding the rightmost branch with an inference in which $\forall x(\forall x Qx \rightarrow Px)_1^2\{\substack{1\\5}\}$ is the principal formula. Then, the instantiation variable u_1^2 is introduced. The connection in question is recreated, but in order to close the other branch as well, it is neccessary to expand it with an inference in which $\exists x(Qx \rightarrow \forall x Px)_6^2\{\substack{12\\53}\}$ is a principal formula. Then, the other connection is also recreated.

Bibliography

[1] Peter B. Andrews. Theorem proving via general matings. *Journal of the Association for Computing Machinery*, 28(2):193–214, 1981.

[2] Roger Antonsen. Free variable sequent calculi. Master's thesis, University of Oslo, Language, Logic and Information, Department of Linguistics, May 2003.

[3] Franz Baader and Wayne Snyder. Unification theory. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 8, pages 445–532. Elsevier Science, 2001.

[4] Matthias Baaz and Christian. G. Fermüller. Non-elementary speedups between different versions of tableaux. In Reiner Hähnle Peter Baumgartner and J. Possega, editors, *4th Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, volume 918 of *LNAI*, pages 217–230. Springer, 1995.

[5] Matthias Baaz and Alexander Leitsch. Complexity of resolution proofs and function introduction. *Annals of Pure and Applied Logic*, 57(3):181–215, 4 June 1992.

[6] Bernhard Beckert. Ein vervollständigungsbasiertes Verfahren zur Behandlung von Gleichheit im Tableaukalkül mit freien Variablen. Diplomarbeit, Fakultät für Informatik, Universität Karlsruhe, July 1993.

[7] Bernhard Beckert. Adding equality to semantic tableaux. In K. Broda, M. D'Agostino, R. Goré, R. Johnson, and S. Reeves, editors, *Proceedings, 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Abingdon*, pages 29–41, Imperial College, London, TR-94/5, 1994.

[8] Bernhard Beckert. Depth-first proof search without backtracking for free-variable clausal tableaux. In P. Baumgartner and H. Zhang, editors, *Third International Workshop on First-Order Theorem Proving (FTP), St. Andrews, Scotland*, pages 44–55, 2000.

[9] Bernhard Beckert and Reiner Hähnle. An improved method for adding equality to free variable semantic tableaux. In D. Kapur, editor, *Proceedings, 11th International Conference on Automated Deduction (CADE), Saratoga Springs, NY*, LNCS 607, pages 507–521. Springer, 1992.

[10] Bernhard Beckert, Reiner Hähnle, and Peter H. Schmitt. The *even more* liberalized δ-rule in free variable semantic tableaux. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Proceedings of the third Kurt Gödel Colloquium KGC'93, Brno, Czech Republic*, volume 713 of *LNCS*, pages 108–119. Springer-Verlag, August 1993.

[11] Wolfgang Bibel. Computationally improved versions of herbrand's theorem. In *Logic Colloquium '81*, pages 11–28. North-Holland, 1982.

[12] Wolfgang Bibel. *Automated Theorem Proving 2. Edition.* Vieweg Verlag, 1987.

[13] Wolfgang Bibel and Elmar Eder. Methods and calculi for deduction. In Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1: Logical Foundations, pages 67–182. Oxford University Press, 1992.

[14] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic.* Cambridge University Press, 2001.

[15] Samuel R. Buss. On Herbrand's theorem. In Daniel Leivant, editor, *Proceedings of the International Workshop on Logic and Computational Complexity (LCC'94)*, volume 960 of *LNCS*, pages 195–209, Berlin, GER, October 1995. Springer.

[16] Domenico Cantone and Marianna Nicolosi Asmundo. A further and effective liberalization of the *delta*-rule in free variable semantic tableaux. In R. Caferra and G. Salzer, editors, *Automated Deduction in Classical and Non-Classical Logics*, LNCS 1761 (LNAI), pages 110–126. Springer, 2000.

[17] Alessandra Carbone. Duplication of directed graphs and exponential blow up of proofs. *Annals of Pure and Applied Logic*, 100(1-3):1–67, 1999.

[18] Anatoli Degtyarev and Andrei Voronkov. Kanger's choices in automated reasoning. In G. Holmström-Hintikka, S. Linström, and R. Sliviski, editors, *Collected papers of Stig Kanger with Essays on his Life and Work*, volume II. Kluwer Academic Publisher, 2001.

[19] Melvin C. Fitting. *First-Order Logic and Automated Theorem Proving*. Graduate Texts in Computer Science. Springer-Verlag, Berlin, 2nd edition, 1996. 1st ed., 1990.

[20] Gerhard Gentzen. Untersuchungen über das logische schliessen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934. English translation [41].

[21] Martin Giese. Integriertes automatisches und interaktives Beweisen: Die Kalkülebene. Diploma Thesis, Fakultät für Informatik, Universität Karlsruhe, June 1998.

[22] Martin Giese. A first-order simplification rule with constraints. In *Proc. Int. Workshop on First-Order Theorem Proving, St. Andrews, Scotland*, 2000.

[23] Martin Giese. Incremental Closure of Free Variable Tableaux. In *Proc. Intl. Joint Conf. on Automated Reasoning, Siena, Italy*, number 2083 in LNCS, pages 545–560. Springer-Verlag, 2001.

[24] Martin Giese. *Proof Search without Backtracking for Free Variable Tableaux*. PhD thesis, Fakultät für Informatik, Universität Karlsruhe, July 2002.

[25] Martin Giese and Wolfgang Ahrendt. Hilbert's ϵ-terms in Automated Theorem Proving. In Neil V. Murray, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, Intl. Conf. (TABLEAUX'99)*, volume 1617 of *LNAI*, pages 171–185. Springer, 1999.

[26] Saul Gorn. Self-annihilating sentences, saul gorn's compendium of rarely used cliches. Technical Report MS-CIS-85-03, Computer and Information Science Department, University of Pennsylvania, January 1985.

[27] Reiner Hähnle. Tableaux and related methods. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 3, pages 100–178. Elsevier Science, 2001.

[28] Reiner Hähnle and Peter H. Schmitt. The liberalized δ-rule in free variable semantic tableaux. *Journal of Automated Reasoning*, 13(2):211–222, October 1994.

[29] David Hilbert and Paul Bernays. *Die Grundlagen der Mathematik II*. Springer-Verlag, Berlin, 2nd edition, 1939.

[30] Herman Ruge Jervell. *Forelesninger i logikk*. Universitetet i Oslo, 1989.

[31] Einar Broch Johnsen and Christoph Lüth. Abstracting theorems for reuse. Submitted for publication, 2003.

[32] Stig Kanger. A simplified proof method for elementary logic. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning 1: Classical Papers on Computational Logic 1957-1966*, pages 364–371. Springer, Berlin, Heidelberg, 1983.

[33] Stephen Cole Kleene. Permutability of inferences in Gentzen's calculi *LK* and *LJ*. Two papers on the predicate calculus. *Memoirs of the American Mathematical Society*, 1952(10):1–26, 1952.

[34] Christoph Kreitz and Jens Otten. Connection-based theorem proving in classical and non-classical logics. *Journal of Universal Computer Science*, 5(3):88–112, 1999.

[35] Christoph Kreitz, Jens Otten, Stephan Schmitt, and Brigitte Pientka. Matrix-based constructive theorem proving. In Steffen Hölldobler, editor, *Intellectics and Computational Logic. Papers in honor of Wolfgang Bibel*, number 19 in Applied Logic Series, pages 189–205. Kluwer, 2000.

[36] Dag Prawitz. An improved proof procedure. *Theoria*, 26:102–139, 1960. Reprinted in [37].

[37] Jörg Siekmann and Graham Wrightson, editors. *Automation of Reasoning: Classical Papers in Computational Logic 1957–1966*, volume 1. Springer-Verlag, 1983.

[38] Raymond M. Smullyan. *First-Order Logic*, volume 43 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer-Verlag, New York, 1968.

[39] Richard Statman. Lower bounds on Herbrand's theorem. *Proceedings of the American Mathematical Society*, 75(1):104–107, 1979.

[40] Gernot Stenz, Wolfgang Ahrendt, and Bernhard Beckert. Proof transformations from search-oriented into interaction-oriented tableau calculi. *Journal of Universal Computer Science*, 5(3):113–134, 1999.

[41] M. E. Szabo, editor. *The Collected Papers of Gerhard Gentzen*. North Holland, Amsterdam, 1969.

[42] Anne S. Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 2 edition, 2000.

[43] Jan van Eijck. Model generation from constrained free variable tableaux. In *Proc. Intl. Joint Conf. on Automated Reasoning, Siena, Italy*, number 2083 in LNCS, pages 150–169. Springer-Verlag, 2001.

[44] Arild Waaler. Connections in nonclassical logics. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 22, pages 1487–1578. Elsevier Science, 2001.

[45] Arild Waaler and Roger Antonsen. A free variable sequent calculus with uniform variable splitting. Accepted for publication at the TABLEAUX 2003 conference, 2003.

[46] Lincoln A. Wallen. *Automated deduction in nonclassical logics*. MIT Press, 1990.

Index

β
- -chain, 87
- -closed, 87
- -closure, 87
- -closure, partial, 87
- -consistent, 87
- -options, 87
- -path, 88
- -path, complete, 88
- -related, 86
- -successor, 87

$\gg$-tree, 47, 100

active, 98
ancestor, 33
- immediate, 33, 46

antecedent, 6
arity, 2
axiom, 12
- σ-, 26

Beethoven, Ludvig van, v
binary, 2
branch
- closed, 13
- open, 13, 40

calculus, 11
closure condition, 11
color, 70, 102
- extension, 74

colored term, 102
complementary β-paths, 88
completeness, 11
conforming
- skeleton, 46, 81

connection, 71
- reduced, 74

consistent, 11
constant symbol, 2
context, 9
copy number, 30, 97
countermodel, 7, 38
- for a skeleton, 40

cycle
- -free, 54
- $\rhd$-, 54
- length, 54

derivation, 10
- closable, 11
- closed, 11
- K-, 11
- length, 16

domain, 5
duality, 87

eigenparameter, 12
- condition, 12

equation set, 71
extra formulae, 98

falsifiable, 7
first-order language, 2
formula
- α-,β-,γ-,δ-, 13
- active, 9
- atomic, 2
- closed, 3
- extra, 9
- L-, 2
- principal, 9

formula occurrence, 97
formula tree, 30
formulae, 97
function symbol, 2

ground
  instantiation term, 24
  substitution, 24
  term, 2

Herbrand universe, 17
  for connections, 71

idempotent, 75
immediate ancestor, 100
index pair, 30, 97
inference, 8
inferences, 100
instantiation
  term, 23
  variable, 23
Instantiation terms, 97
instantiation variables, 97
interpretation function, 5

LK-like, 50, 82
logical consequence, 6
logical symbols, 2

model, 6
multiset, 6

n-ary, 2
name, 5

occurrence
  active formula, 9
  extra formula, 9
  principal formula, 9
occurrence number, 30, 97

permutation, 50
  variant, 50
polarity, 30
predicate symbol, 2
principal formula, 98
principal type, 30
projective, 82
proof, 11
provable, 11

quantification variables, 97

reduction ordering, 54
root sequent, 98
rule, 8
  application, 8
  one-premiss, 8
  two-premiss, 8

satisfiable, 6
schema, 8
sequent, 6
  root, 11
sequents, 98
skeleton, 24, 98
  balanced, 49
  complete, 87
  variable-pure, 51
  variable-sharing, 51
Skolem function, 23
Skolem function symbols, 97
Skolem term, 23
soundness, 11
spanning, 71
splitting set, 67, 97
strategy, 18
  fair, 18
structure, 5
subformula, 33
  immediate, 33
  occurrence, 33
subsequent, 6
substitution, 4
  closing, 24, 71
  composition, 4
  for connections, 71
  for instantiation terms, 24
succedent, 6
support, 4

term
  colored, 71
  L-, 2

unary, 2
unifier, 24

valid
  formula, 6
  sequent, 7, 38
variable
  assignment, 37
  bound, 3
  colored, 71
  free occurrence, 3
  quantification, 23
  rigid, 40
  universal, 66

well-founded, 55